Universiteit Leiden
The Netherlands

Exploring deep learning for intelligent image retrieval

Chen, W.

Citation

Chen, W. (2021, October 13). *Exploring deep learning for intelligent image retrieval*. Retrieved from https://hdl.handle.net/1887/3217054

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3217054

Note: To cite this publication please use the final published version (if applicable).

Exploring Deep Learning for Intelligent Image Retrieval

Wei Chen

Copyright © 2021 Wei Chen, All Rights Reserved

ISBN 978-94-6419-293-3

Printed by Gildeprint, The Netherlands

Cover design: Zhihan Zhao, Wei Chen

Exploring Deep Learning for Intelligent Image Retrieval

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op woensdag 13 oktober 2021
klokke 15.00 uur

door

Wei Chen

geboren te Guizhou, China
in 1991

Promotiecommissie

Promotors:	Prof. dr. M.S. Lew
	Prof. dr. A. Plaat
Overige leden:	Prof. dr. T.S. Chua (National University of Singapore)
	Prof. dr. B.P.F. Lelieveldt
	Prof. dr. T.H.W. Bäck
	Dr. E.M. Bakker
	Dr. K.J. Wolstencroft

Wei Chen was financially supported through the China Scholarship Council (CSC) to participate in the PhD programme of Leiden University. Grant number 201703170183.

The research in this thesis was performed at the LIACS MediaLab, Leiden University, The Netherlands, and we would like to thank the NVIDIA Corporation for the donation of GPU cards.

To my family

Contents

1 Introduction **1**
1.1 Background . . . 2
1.2 Research Questions and Contributions . . . 3
1.3 Thesis Overview . . . 5

2 A Comprehensive Review of Deep Image Retrieval **9**
2.1 Introduction . . . 10
2.1.1 Summary of progress since 2012 . . . 12
2.1.2 Key challenges . . . 14
2.2 Deep Convolutional Neural Networks . . . 15
2.2.1 A brief introduction to deep learning . . . 15
2.2.2 Popular backbone DCNN architectures . . . 16
2.3 Retrieval with Off-the-Shelf DCNN Models . . . 17
2.3.1 Deep feature extraction . . . 18
2.3.1.1 Network feedforward scheme . . . 18
2.3.1.2 Deep feature selection . . . 19
2.3.1.3 Feature fusion strategy . . . 21
2.3.2 Deep feature enhancement . . . 23
2.3.2.1 Feature aggregation . . . 23
2.3.2.2 Feature embedding . . . 23
2.3.2.3 Attention mechanisms . . . 26
2.3.2.4 Deep hash embedding . . . 27
2.4 Retrieval via Learning DCNN Representations . . . 28
2.4.1 Supervised fine-tuning . . . 29
2.4.1.1 Classification-based fine-tuning . . . 29
2.4.1.2 Verification-based fine-tuning . . . 30
2.4.2 Unsupervised fine-tuning . . . 34
2.4.2.1 Mining samples with manifold learning . . . 35
2.4.2.2 AutoEncoder-based frameworks . . . 36
2.5 State of the Art Performance . . . 38
2.5.1 Datasets . . . 38
2.5.2 Evaluation metrics . . . 38
2.5.3 Performance comparison and analysis . . . 39

2.6 Chapter Conclusions 43

3 Domain Uncertainty based on Information Theory for Cross-modal Hash Retrieval 47

3.1 Introduction 48
3.2 Cross-modal Hash Learning 49
3.3 Domain Uncertainty Measurement via Information Theory 50
3.3.1 Information theory and domain uncertainty 50
3.3.2 Multi-level feature preserving 51
3.3.3 Classification-based objective function 53
3.4 Experiments and Evaluations 54
3.4.1 Implementation details 54
3.4.2 Datasets 54
3.4.3 Performance and evaluation 54
3.5 Chapter Conclusions 57

4 Integrating Information Theory and Adversarial Learning for Cross-modal Retrieval 59

4.1 Introduction 60
4.2 Related Work 61
4.2.1 Cross-modal representation learning and matching 61
4.2.2 Adversarial learning for cross-modal retrieval 62
4.2.3 Information-theoretical feature learning 63
4.3 Proposed Approach 64
4.3.1 Problem formulation 64
4.3.2 Integrating information theory & adversarial learning 64
4.3.2.1 Information entropy and modality uncertainty 64
4.3.2.2 Adversarial learning and information entropy 65
4.3.3 KL-divergence for cross-modal feature projection 66
4.4 Implementation and optimization 67
4.4.1 Combining information theory & adversarial learning 67
4.4.2 KL-divergence for similarity preserving 69
4.4.3 Instance label classification 70
4.4.3.1 Categorical cross-entropy loss 70
4.4.3.2 KL-divergence for data imbalance 70
4.4.4 Bi-directional triplet constraint 71
4.5 Experiments 73
4.5.1 Datasets and settings 73
4.5.2 Performance evaluation 74
4.5.2.1 Results on the Flickr30K and MS-COCO datasets . . 74
4.5.2.2 Results on CUHK-PEDES dataset 75
4.5.2.3 Results on Flickr8K dataset 76
4.5.3 Ablation study 76

4.5.3.1 Analysis of KL-divergence for data imbalance 77
4.5.3.2 Analysis of KL-divergence for cross-modal feature projection 77
4.5.3.3 Analysis of adversary combining 77
4.5.3.4 Analysis of temperature τ 78
4.5.3.5 Distribution visualization 78
4.5.3.6 Analysis of complexity and stability 80
4.5.4 Further exploring 81
4.6 Chapter Conclusions 83

5 On the Exploration of Incremental Learning for Fine-grained Image Retrieval 85
5.1 Introduction 86
5.2 Related Work 87
5.3 Proposed Approach 87
5.3.1 Semantic preserving loss 88
5.3.2 Knowledge distillation loss 89
5.3.3 Maximum mean discrepancy loss 89
5.4 Experiments 91
5.4.1 Datasets and experimental settings 91
5.4.2 One-step incremental learning for FGIR 91
5.4.3 Multi-step incremental learning for FGIR 94
5.4.4 Validation with image classification 97
5.4.5 Training time comparison 98
5.4.6 Components analysis 98
5.5 Chapter Conclusions 98

6 Feature Estimations based Correlation Distillation for Incremental Image Retrieval 101
6.1 Introduction 102
6.2 Related Work 103
6.3 Correlations Distillation for Incremental Image Retrieval 104
6.3.1 Problem formulation 104
6.3.2 Correlations distillation for one-task scenario 106
6.3.3 Feature estimation for multi-task scenario 106
6.4 Experiments 110
6.4.1 Datasets and experimental setup 110
6.4.2 One-task scenario evaluation 111
6.4.3 Multi-task scenario evaluation 114
6.4.4 Ablation study 116
6.4.5 Retrieval visualization 120
6.5 Chapter Conclusions 121

7 Lifelong Image Retrieval via Dual Knowledge Distillation 125
7.1 Introduction . . . 126
7.2 Related Work . . . 127
7.3 The Lifelong Image Retrieval Problem . . . 128
7.4 Dual Knowledge Distillation . . . 129
7.4.1 Knowledge distillation by frozen teacher . . . 129
7.4.2 Representative data generation . . . 130
7.4.3 Self-motivated learning on the mixed data . . . 131
7.4.4 Auxiliary distillation by on-the-fly teacher . . . 131
7.5 Experiments . . . 132
7.5.1 Dataset splits . . . 132
7.5.2 Training details . . . 133
7.5.3 Performance evaluation . . . 133
7.5.4 Further explorations . . . 141
7.6 Chapter Conclusions . . . 142

8 New Ideas and Trends in Deep Multimodal Content Understanding 143
8.1 Introduction . . . 144
8.2 Multimodal Applications . . . 145
8.2.1 Uni-directional applications . . . 145
8.2.2 Bi-directional applications . . . 146
8.3 Recent Advances in Content Understanding . . . 148
8.3.1 Deep multimodal structures . . . 148
8.3.2 Multimodal feature extraction . . . 151
8.3.3 Common latent space learning . . . 156
8.4 Results and Discussions . . . 162
8.5 Chapter Conclusions . . . 168

9 Conclusions 169
9.1 Limitations and Possible Solutions . . . 172
9.2 Future Research Directions . . . 172

Bibliography 175

List of Abbreviations 195

English Summary 197

Nederlandse Samenvatting 199

Curriculum Vitae 205

Chapter 1

Introduction

1.1 Background

The use of images to convey information is playing a crucial role in our daily life in diverse areas such as medicine, journalism, advertising, design, education, and entertainment. The rapid growth of images has been facilitated by numerous technologies ranging from fast Internet to digital cameras and smartphones to inexpensive solid state storage. For example, in 2020, the number of smartphone users worldwide surpasses 3.6 billion and is forecast to further grow by several hundred million in the next few years. It is straightforward to capture and store the visual imagery. However, searching or retrieving the image is difficult because computers do not understand the relationship between the image pixel representations and high level human concepts. Image retrieval has become a major research area worldwide. Given a query image that describes the user's needs, image retrieval is the process of searching for semantically matched or similar images in a large image dataset by analyzing their visual content, for example, a zoologist would look for the photograph of a particular animal, and so on.

To enable accurate and efficient retrieval in massive image collections, compact and rich image feature representations are at the core of successful image retrieval. In the past two decades, remarkable progress has been made, focusing more on using primitive colour features, texture features, and shape features [1, 2, 3]. These low-level features heavily depend on hand-engineered feature descriptors such as Scale-Invariant Feature Transform (SIFT) [4], Speeded Up Robust Features (SURF) [5], and Histogram of Oriented Gradients (HOG) [6]), middle-level representations such as bag of words (BoW) [7] and Fisher Vector (FV) [8, 9]. However, the low-level image features do not have sufficient discriminatory power to effectively bridge the semantic gap, which characterizes the difference between the high-level concepts of humans and the low-level features typically derived from images [10]. In recent years, deep learning, particularly the influential ImageNet [11] and the Deep Convolutional Neural Network (DCNN) AlexNet [12] enables learning powerful feature representations with multiple levels of abstraction directly from data. Deep learning techniques have attracted enormous attention and have brought about considerable breakthroughs in various computer vision tasks including image classification [12, 13, 14], object detection [15, 16, 17, 18], semantic segmentation [19, 20, 21], and image generation [22, 23, 24]. This has also directly impacted the field of image retrieval [25, 26, 27, 28, 29, 30, 31, 32]. DCNNs can learn high-level semantic-aware features to perform retrieval tasks. For instance, current literature may directly use a pre-trained deep model to extract features for input images. Likewise, visual features can be captured from the sub-patches of images to satisfy instance-level image retrieval. Different layers (*e.g.* convolutional layers and fully-connected layers) can capture different levels of features: global features (by fully-connected layers) and local features (by convolutional layers). Model-level and layer-level feature fusion schemes are further studied to incorporate multi-scale deep features by using

methods such as concatenation and weighted averaging. Despite these strategies to improve the performance of image retrieval, there exist many other problems to be addressed, which is the core of this thesis.

1.2 Research Questions and Contributions

Semantic information that helps us understand the world usually comes from different sensory modalities. In the real world, we can express the same concept using different ways: for example by using writing text or by taking a picture. People can search for images of interest by submitting any media content at hand (*e.g.* a word, a phrase, or a sentence) as the query item. Cross-modal retrieval, as a natural search method becomes increasingly important to augment image retrieval. For cross-modal retrieval, although raw data from two modalities (*e.g.* an image and a text) have similar semantic concepts, feature vectors extracted from these data are distributed in different spaces so that their semantics are not well associated. Therefore, these vectors are not directly comparable, leading to inconsistent distributions. This is what is referred to as a heterogeneity gap. The challenge of performing cross-modal retrieval lies in how to measure the semantic similarity, which is defined as a metric to quantify the likeness of two items. In the context of cross-modal retrieval, these items (image and text data) are from different modalities. In terms of this challenge, we come to our first **Research Question 1** (**RQ 1**): **How to learn a commonly shared embedding space for visual modality and textual modality to reduce the heterogeneity gap?** Focusing on the **RQ 1**, we explore cross-modal retrieval by using information entropy to measure the domain uncertainty of the shared space for visual and textual modalities. Furthermore, we combine information theory and adversarial learning into an end-to-end framework. This work is the first to explore information theory in reducing the heterogeneity gap for cross-modal retrieval, a method that is beneficial for constructing a shared space for learning commonalities between cross-modal features. In addition, we introduce a regularization term based on Kullback-Leibler (KL) divergence [33] with temperature scaling to address the issue of data imbalance. Our contributions are based on the following publications [34, 35]:

> **Chen, W.**, Pu, N., Liu, Y., Bakker, E. M., and Lew, M. S., "Domain Uncertainty Based On Information Theory for Cross-Modal Hash Retrieval." IEEE International Conference on Multimedia and Expo, 2019, pp. 43-48.
>
> **Chen, W.**, Liu, Y., Bakker, E. M., and Lew, M. S., "Integrating Information Theory and Adversarial Learning for Cross-modal Retrieval." Pattern Recognition, 2021, 117, pp. 107983.

The existing fine-grained datasets for image retrieval have limited categories to support a retrieval system. As noted, the wide popularity of mobile devices make large

image collections available. However, the deep models are only trained and validated on these limited image categories. The retrieval ability of deep models is limited to the existing categories, but cannot be extended to some of new incoming categories. Unlike the continuous learning process of human beings, deep neural networks suffer from a catastrophic forgetting problem [36], a phenomenon that occurs when a network is trained successively on a series of new data and the learning of this data degrades the performance on previous data. To satisfy domain context constraints, the deep models are required to learn on a series of categories sequentially. Thus we come to our second **RQ 2: What kind of knowledge is more beneficial, for a deep model to learn fine-grained categories incrementally, in order to reduce catastrophic forgetting of previous data?** We extend fine-grained image retrieval in the context of incremental learning and explore this question in the following publication [37]:

> **Chen, W.**, Liu, Y., Wang, W., Tuytelaars, T., Bakker, E. M., and Lew, M. S., "On the Exploration of Incremental Learning for Fine-grained Image Retrieval." The British Machine Vision Conference (BMVC), 2020, pp. 1-10.

To the best of our knowledge, this is the first work to study this problem. Furthermore, incremental learning has been employed to make deep models learn on old data and newly added data successively. However, as incremental learning proceeds, each training session produces a specific model. Saving this stream of models is memory-consuming. Thus, we come to our third **RQ 3: How to utilize the model stream in incremental learning to transfer previously learned information to the deep model trained on the new data?** We investigate this question by proposing a feature estimation method for incremental fine-grained image retrieval, which is based on the following publication [38]:

> **Chen, W.**, Liu, Y., Pu, N., Wang, W., Liu L., and Lew, M.S., "Feature Estimations based Correlation Distillation for Incremental Image Retrieval." IEEE Transactions on Multimedia, 2021.

In general, deep models are trained and validated on fixed or stationary datasets. Exploring incremental learning by adding new fine-grained categories is still far from realizing the model's continuous retrieval ability because the images in old categories and new incoming categories are semantically similar. However, the images added to the fixed datasets may have different semantic contents, for instance when a model trained on a vehicle dataset is transferred to learn a new dataset which includes different breeds of flowers. For the context of incremental learning, the noted semantic shifts between different datasets (*e.g.* flower and car images) make the problem of minimizing the forgetting ratio more difficult when training deep models in a lifelong manner. In this case, we consider the fourth **RQ 4: How to perform lifelong image retrieval on different training datasets by reducing**

the impact of semantic shifts? We investigate this question by proposing a dual knowledge distillation framework, which is based on the submitted manuscript:

> **Chen, W.**, Pu, N., Liu, Y. , Lao, M., Wang, W., Bakker, E. M., Liu L., Tuytelaars, T., and Lew, M.S., "Lifelong Image Retrieval via Dual Knowledge Distillation." submitted to Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI) (*under review*), 2021.

1.3 Thesis Overview

The contents of this thesis is based on the articles where I have been the primary author. To organize this thesis for a better understanding, in **Chapter 2** we first present a comprehensive review of intelligent image retrieval via deep learning. It includes the main challenges of intelligent image retrieval, the categorization for retrieval methodologies, the popular convolutional neural networks for image retrieval, *etc.* This chapter aims at giving a global view for intelligent image retrieval, and is based on the submitted manuscript:

> **Chen, W.**, Liu, Y., Wang, W., Bakker, E. M., Georgiou T., Fieguth P., Liu L., and Lew, M. S., "Deep Image Retrieval: A Survey." submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (*Major revision*), 2021.

In Chapter 3 and Chapter 4, we target **RQ1** and present related work on cross-modal retrieval. We focus on cross-modal hash retrieval in **Chapter 3** and propose using information theory for measuring the domain uncertainty of binary codes in the shared feature space for image and text. In **Chapter 4**, we further combine information theory in reducing the heterogeneity gap for cross-modal retrieval. This method is beneficial for constructing a shared space for further learning commonalities between cross-modal features.

In **Chapter 5**, we target **RQ2** and explore image retrieval in the context of incremental learning. We focus on expanding the continuous retrieval ability of deep networks, and explore using feature correlations between deep features as knowledge to transfer from the teacher model to the student model. This research is limited to fine-grained datasets.

In **Chapter 6**, we target **RQ3** and explore how to utilize the model stream in incremental fine-grained image retrieval. We propose a feature estimation method to transfer more previous knowledge to further minimize the forgetting ratio on previous old data. The benefit of the feature estimation method is that the streams of deep models trained for the previous tasks are unnecessarily saved. We demonstrate the effectiveness of the proposed method on fine-grained datasets.

In **Chapter 7**, we target **RQ4** and explore lifelong image retrieval on different datasets. Compared to the proposed methods in Chapter 5 and Chapter 6, lifelong

image retrieval is important to numerous practical retrieval applications. To train a deep model on different datasets sequentially, the semantic shifts between the training data (*e.g.* from flower images to vehicle images) will make the problem of minimizing the forgetting ratio difficult. Moreover, prior works for image retrieval mainly improve the generalization ability of the deep model, but did not consider reducing forgetting simultaneously. To address these challenges, we propose a dual knowledge distillation framework and utilize the stored statistics in the BatchNorm layers of a frozen teacher model, which can minimize the forgetting ratio on the old tasks and simultaneously improve generalization on the new task.

In **Chapter 8**, we present recent new ideas and trends for multimodal content understanding. These methods can be used for intelligent image retrieval to seek performance improvement. This chapter is based on the following publication [39]:

> **Chen, W.**, Wang, W., Liu, L., and Lew, M.S., "New Ideas and Trends in Deep Multimodal Content Understanding: A Review." Neurocomputing, 2020, pp. 195-215.

Finally, in **Chapter 9** we conclude this thesis and reflect on future research directions. In addition, this thesis has been inspired by the insights and experiences from the related works in the following publications during my PhD studies:

- Liu, Y., **Chen, W.**, Liu, L., and Lew, M. S., "SwapGAN: A Multi-stage Generative Approach for Person-to-Person Fashion Style Transfer." IEEE Transactions on Multimedia, 2019, 21(9), pp. 2209-2222.
- Pu, N., **Chen, W.**, Liu, Y., Bakker, E. M., and Lew, M. S. "Dual Gaussian-based Variational Subspace Disentanglement for Visible-Infrared Person Re-Identification." ACM International Conference on Multimedia, 2020, pp. 2149-2158.
- Pu, N., **Chen, W.**, Liu, Y., Bakker, E. M., and Lew, M. S. "Lifelong Person Re-Identification via Adaptive Knowledge Accumulation." IEEE International Conference on Computer Vision and Pattern Recognition, 2021.
- Liu, Y., Guo, Y., **Chen, W.**, and Lew, M. S., "An Extensive Study of Cycle-Consistent Generative Networks for Image-to-Image Translation." IEEE International Conference on Pattern Recognition, 2018, pp. 219-224.
- Georgiou T, Liu Y, **Chen W**, Lew M. S. "A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision." International Journal of Multimedia Information Retrieval, 2020, 9(3), pp. 135-170.
- Lao, M., Guo, Y., Pu, N., **Chen, W.**, Liu, Y., and Lew, M. S. "Multi-stage hybrid embedding fusion network for visual question answering." Neurocomputing, 2021, 423, pp. 541-550.

- Lao, M., Guo, Y., Liu, Y., **Chen, W.**, Pu, N., and Lew, M. S. "From Superficial to Deep: Language Bias driven Curriculum Learning for Visual Question Answering." ACM International Conference on Multimedia, 2021.
- Georgiou, T., Schmitt, S., Bäck, T., **Chen, W.**, and Lew, M. S. "Norm Loss: An efficient yet effective regularization method for deep neural networks." IEEE International Conference on Pattern Recognition, 2020, pp. 8812-8818.
- Georgiou, T., Schmitt, S., Bäck, T., Pu, N., **Chen, W.**, and Lew, M. S. (2021). "Comparison of deep learning and hand crafted features for mining simulation data." IEEE International Conference on Pattern Recognition, 2020, pp. 1-8.

Chapter 2

A Comprehensive Review of Deep Image Retrieval

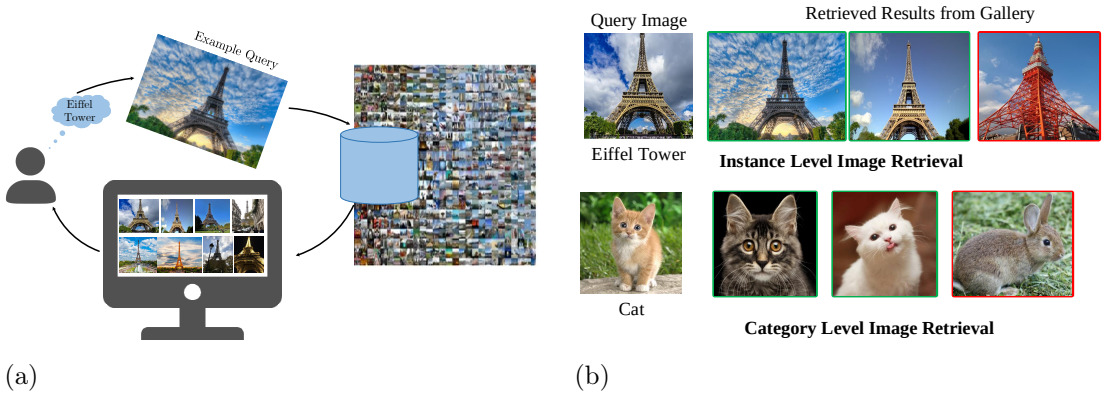
In recent years a vast amount of visual content has been generated and shared from various fields, such as social media platforms, medical images, and robotics. This abundance of content creation and sharing has introduced new challenges. In particular, searching databases for similar content, *i.e.* content based image retrieval (CBIR), is a long-established research area, and more efficient and accurate methods are needed for real time retrieval. Artificial intelligence has made progress in CBIR and has significantly facilitated the process of intelligent search. In this chapter, we organize and review recent CBIR works that are developed based on deep learning algorithms and techniques, including insights and techniques from recent papers. We identify and present the commonly-used benchmarks and evaluation methods used in the field. We collect common challenges and propose promising future directions. More specifically, we focus on image retrieval with deep learning and organize the state of the art methods according to the types of deep network structure, deep features, feature enhancement methods, and network fine-tuning strategies. Our survey considers a wide variety of recent methods, aiming to promote a global view of the field of instance-based CBIR.

Keywords

Content based image retrieval, Deep learning, Convolutional neural networks, Literature review

This chapter is based on the following publication:

- Chen, W., Liu, Y., Wang, W., Bakker, E., Georgiou T., Fieguth P., Liu L., and Lew, M.S., "Deep Image Retrieval: A Survey." submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (*major revision*), 2021.



(a) (b)

Figure 2.1: Illustration of (a) the CBIR process and (b) categorization. The images in green frame are retrieved correctly, while the ones in red frame are matched incorrectly.

2.1 Introduction

Content based image retrieval (CBIR) is the problem of searching for semantically matched or similar images in a large image gallery by analyzing their visual content, given a query image that describes the user's needs. CBIR has been a longstanding research topic in the computer vision and multimedia community [1, 40]. With the present, exponentially increasing, amount of image and video data, the development of appropriate information systems that efficiently manage such large image collections is of utmost importance, with image searching being one of the most indispensable techniques.

A broad categorization of CBIR methodologies depends on the level of retrieval, *i.e.* instance level and category level. In instance level image retrieval, a query image of a particular object or scene (*e.g.* the Eiffel Tower) is given and the goal is to find images containing the same object or scene that may be captured under different conditions [3, 25]. In contrast, the goal of category level retrieval is to find images of the same class as the query (*e.g.* dogs, cars, *etc.*). Instance level retrieval is more challenging and promising as it satisfies specific objectives for many applications. Notice that we limit the focus of this chapter to instance-level image retrieval and in the following, if not further specified, "image retrieval" and "instance retrieval" are considered equivalent and will be used interchangeably.

Finding a desired image can require a search among thousands, millions, or even billions of images. Hence, searching efficiently is as critical as searching accurately, to which continued efforts have been devoted [3, 25, 26, 41]. To enable accurate and efficient retrieval of massive image collections, *compact yet rich feature representations* are at the core of CBIR.

In the past two decades, remarkable progress has been made in image feature representations, which mainly consist of two important periods: feature engineering and feature learning (particularly deep learning). In the feature engineering era (*i.e.*

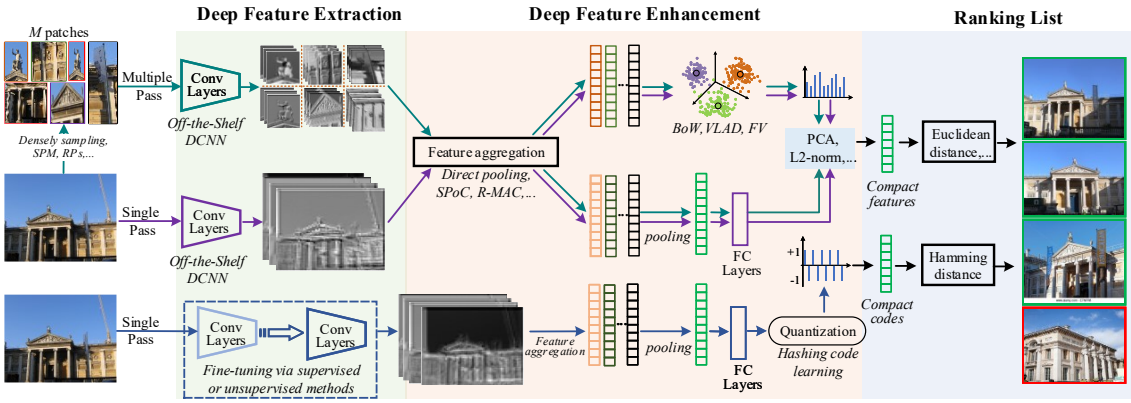


Figure 2.2: In deep image retrieval, feature embedding and aggregation methods are used to enhance the discrimination of deep features. Similarity is measured on these enhanced features using Euclidean or Hamming distances.

pre-deep learning), the field was dominated by milestone hand-engineered feature descriptors, such as the Scale-Invariant Feature Transform (SIFT) [4]. The feature learning stage, the deep learning era since 2012, begins with artificial neural networks, particularly the breakthrough ImageNet and the Deep Convolutional Neural Network (DCNN) AlexNet [12]. Since then, deep learning has impacted a broad range of research areas, since DCNNs can learn powerful feature representations with multiple levels of abstraction directly from data. Deep learning techniques have attracted enormous attention and have brought about considerable breakthroughs in many computer vision tasks, including image classification [12, 13, 14], object detection [17], and image retrieval [26, 27, 42].

Excellent surveys for traditional image retrieval can be found in [1, 3, 40]. This chapter, in contrast, focuses on deep learning based methods. Deep learning for image retrieval is comprised of the essential stages shown in Figure 2.2 and various methods, focusing on one or more stages, have been proposed to improve retrieval accuracy and efficiency. In this chapter, we include comprehensive details about these methods, including feature fusion methods and network fine-tuning strategies *etc*, motivated by the following questions that have been driving research in this domain:

1. *By using off-the-shelf models only, how do deep features outperform hand-crafted features?*

2. *In case of domain shifts across training datasets, how can we adapt off-the-shelf models to maintain or even improve retrieval performance?*

3. *Since deep features are generally high-dimensional, how can we effectively utilize them to perform efficient image retrieval, especially for large-scale datasets?*

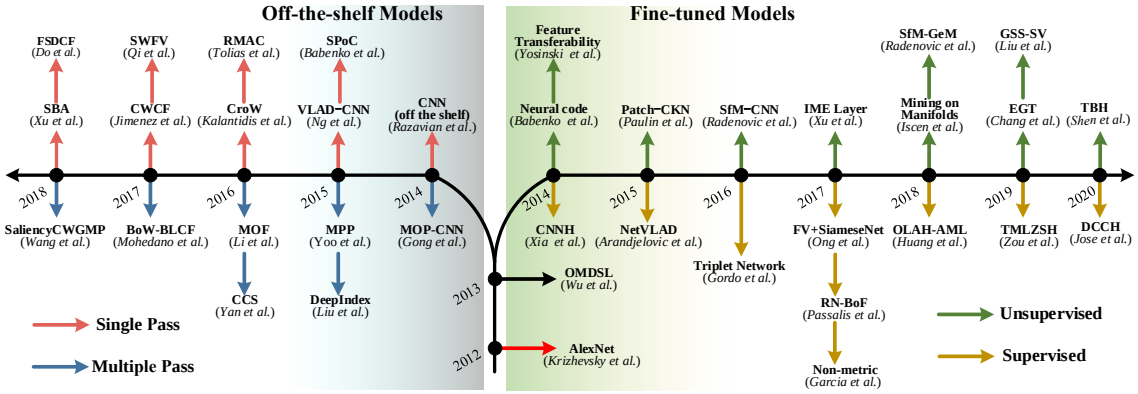


Figure 2.3: Representative methods in deep image retrieval, which are most fundamentally categorized according to whether the DCNN parameters are updated [43]. Off-the-shelf models (left) have model parameters which are not further updated or tuned when extracting features for image retrieval. The relevant methods focus on improving representations quality either by feature enhancement [26, 45, 46, 47] when using single pass schemes or by extracting representations for image patches [48] when using multiple pass schemes. In contrast, in fine-tuned models (right) the model parameters are updated for the features to be fine-tuned towards the retrieval task and addresses the issue of domain shifts. The fine-tuning may be supervised [49, 50, 51, 52, 53, 54, 55] or unsupervised [32, 56, 57, 58, 59, 60]. See Sections 2.3 and 2.4 for details.

2.1.1 Summary of progress since 2012

After a highly successful image classification implementation based on AlexNet [12], significant exploration of DCNNs for retrieval tasks has been undertaken, broadly along the lines of the preceding three questions just identified, above. That is, the DCNN methods are divided into (1) off-the-shelf and (2) fine-tuned models, as shown in Figure 2.3, with parallel work on (3) effective features. Whether a DCNN is considered off-the-shelf or fine-tuned depends on whether the DCNN parameters are updated [43] or are based on DCNNs with fixed parameters [29, 43, 44]. Regarding how to use the features effectively, researchers have proposed encoding and aggregation methods, such as R-MAC [31], CroW [26], and SPoC [25].

Recent progress for improving image retrieval can be categorized into network-level and feature-level perspectives, for which a detailed sub-categorization is shown in Figure 2.4. The network-level perspective includes network architecture improvement and network fine-tuning strategies. The feature-level perspective includes feature extraction and feature enhancement methods. Broadly this chapter will examine the four areas outlined as follows:

a. Improvements in network architectures (section 2.2.2)

Using stacked linear filters (*e.g.* convolution) and non-linear activation functions (ReLU, *etc.*), deep networks with different depths obtain features at different levels. Deeper networks with more layers provide a more powerful learning capacity so as to

- Deep learning for image retrieval
  - **Improvement in deep network architectures (section 2.2.2)**
    - Deepen networks: AlexNet [12], VGG [61], ResNet [13], *etc*
    - Widen networks: GoogLeNet [62], DenseNet [14], *etc*
  - Retrieval with off-the-shelf DCNN models (section 2.3)
    - **Deep feature extraction (section 2.3.1)**
      - Network feedforward scheme (section 2.3.1.1)
        - Single feedforward pass: MAC [63], R-MAC [31]
        - Multiple feedforward pass: SPM [48], RPNs [54]
      - Deep feature selection (section 2.3.1.2)
        - Fully-connected layer: layer concatenation [64]
        - Convolutional layer: SPoC [25], CroW [26]
      - Feature fusion strategy (section 2.3.1.3)
        - Layer-level fusion: MoF [65], MOP [29]
        - Model-level fusion: ConvNet fusion [61]
    - **Deep feature enhancement (section 2.3.2)**
      - Feature aggregation (section 2.3.2.1)
      - Feature embedding (section 2.3.2.2)
      - Attention mechanism (section 2.3.2.3)
        - Non-parameteric: SPoC [25], TSWVF [66]
        - Parameteric: DeepFixNet+SAM [67, 68]
      - Deep hash embedding (section 2.3.2.4)
        - Supervised hashing: Metric Learning [51]
        - Unsupervised hashing: KNN [69], k-means [70]
  - **Retrieval via learning DCNN representations (section 2.4)**
    - Supervised fine-tuning (section 2.4.1)
      - Classification-based fine-tuning (section 2.4.1.1)
      - Verification-based fine-tuning (section 2.4.1.2)
        - Transformation matrix: Non-metric [52]
        - Siamese networks: [53, 71]
        - Triplet networks: [53, 71]
    - Unsupervised fine-tuning (section 2.4.2)
      - Manifold learning sample mining: Diffusion Net [58]
      - AutoEncoder-based fine-tuning: KNN [72], GANs [60]

Figure 2.4: This chapter is organized around four key aspects in deep image retrieval, shown in boldface.

extract high-level abstract and semantic-aware features [13, 61]. It is also possible to concatenate multi-scale features in parallel, such as the Inception module in GoogLeNet [62], which we refer to as widening.

b. Deep feature extraction (section 2.3.1)

Neurons of FC layers and convolutional layers have different receptive fields, thus providing three ways to extract features: local features from convolutional layers [25, 31], global features from FC layers [48, 73] and fusions of two kinds of features [74, 75]; the fusion scheme includes layer-level and model-level methods. Deep features can be extracted from the whole image or from image patches, which corresponds to single pass and multiple pass feedforward schemes, respectively.

c. Deep feature enhancement (section 2.3.2)

Feature enhancement is used to improve feature's discriminative ability. Directly,

aggregate features can be trained simultaneously with deep networks [76]; alternatively, feature embedding methods including BoW [7], VLAD [28], and FV [8] embed local features into global ones. These methods are trained with networks separately (codebook-based) or jointly (codebook-free). Further, hashing methods [77] encode the real-valued features into binary codes to improve retrieval efficiency. The feature enhancement strategy significantly influences the efficiency of image retrieval.

d. Network fine-tuning for learning representations (section 2.4)

Deep networks pre-trained on source datasets for image classification are transferred to new datasets for retrieval tasks. However, the retrieval performance is influenced by the domain shifts between the datasets. Therefore, it is necessary to fine-tune the deep networks to the specific domain [50, 70, 78], which can be realized by using supervised fine-tuning methods. However in most cases image labeling or annotation is time-consuming and difficult, so it is necessary to develop unsupervised methods for network fine-tuning.

2.1.2 Key challenges

Deep learning has been successful in learning powerful features. Nevertheless, several significant challenges remain with regards to

1. *reducing the semantic gap*,
2. *improving retrieval scalability*, and
3. *balancing retrieval accuracy and efficiency.*

We finish the introduction to this chapter with a brief overview of each of these challenges:

1. Reducing the semantic gap: The semantic gap characterizes the difference, in any application, between the high-level concepts of humans and the low-level features typically derived from images [10]. There is significant interest in learning deep features which are higher-level and semantic-aware, to better preserve the similarities of images [10]. In the past few years, various learning strategies, including feature fusion [29, 65] and feature enhancement methods [25, 31, 66] have been introduced into image retrieval. However, this area remains a major challenge and continues to require significant effort.

2. Improving retrieval scalability: The tremendous numbers and diversity of datasets lead to domain shifts for which existing retrieval systems may not be suited [3]. Currently available deep networks are initially trained for classification tasks, which leads to a challenge in extracting features. Since such features are less scalable and perform comparatively poorly on the target retrieval datasets, so network fine-tuning on retrieval datasets is crucial for mitigating this challenge. The current

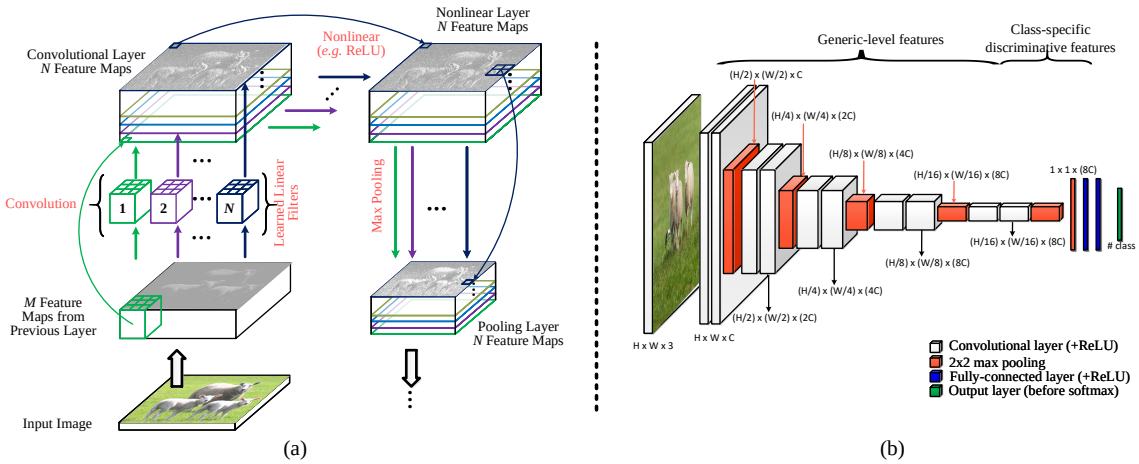


Figure 2.5: (a) Illustration of three operations that are repeatedly applied by a typical CNN [79]. (b) Generic framework of CNN.

dilemma is that the increase in retrieval datasets raises the difficulty of annotation, making the development of unsupervised fine-tuning methods a priority.

3. Balancing retrieval accuracy and efficiency: Deep features are usually high dimensional and contain more semantic-aware information to support higher accuracy, yet this higher accuracy is often at the expense of efficiency. Feature enhancement methods, like hash learning, are one way to tackle this issue [50, 77], however hashing learning needs to carefully consider the loss function design, such as quantization loss [41], to obtain optimal codes for high retrieval accuracy.

2.2 Deep Convolutional Neural Networks

2.2.1 A brief introduction to deep learning

Deep learning depends on neural networks to learn features. Deep neural networks have various variants. Among them, convolutional neural networks (CNNs) are used for vision tasks. There are three types of layer in CNNs: convolutional layer, pooling layer, and fully-connected layer [79]. The convolutional layer plays a vital role in the way CNNs work, emphasizing the use of shared and learnable 2D linear filters. As illustrated in Figure 2.5(a), when a filter glides through the M feature maps from the previous layer $l-1$ each time, the outputs of the convolutions for the next layer l are calculated with its parameters $\boldsymbol{\theta}$, that includes weights $\boldsymbol{w}$ and bias $\boldsymbol{b}$:

$$\boldsymbol{x}^l = \sum_{i=1}^{M^{l-1}} \left(\boldsymbol{w}_i \boldsymbol{x}_i^{l-1} \; + \; \boldsymbol{b} \right) \tag{2.1}$$

It is important to impose a non-linear activation function $\sigma(\cdot)$ (*e.g.* ReLU) on the feature maps $\boldsymbol{x}^l$. Finally, the outputs of non-linear function is stored as inputs

for the next layer l. Usually, the number of filters applied in the previous layer determines the number of produced feature maps in the next layer. As illustrated in Figure 2.5(a), the N filters produce N feature maps.

Finally, the difference between the predictive logits of the classifier and the ground-truth label is used to compute gradients to train the network. Take supervised training as an example, a ground-truth label y^j is assigned to an input x^j, the loss function for network $f(\cdot, \boldsymbol{\theta})$ can then be formulated as:

$$J(\boldsymbol{\theta}) \;=\; \textstyle\sum_j L\big(f(x^j; \boldsymbol{\theta}),\; y^j\big) \tag{2.2}$$

During training, the gradients are computed according to the loss function $J(\boldsymbol{\theta})$ and are back-propagated to $f(\cdot, \boldsymbol{\theta})$, aiming at learning the optimal parameters $\boldsymbol{\theta}^*$:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{argmin}\; J(\boldsymbol{\theta}) \tag{2.3}$$

A convolutional layer represents local feature learning and yields generic features [44], as shown in Figure 2.5(b). Specifically, the first convolutional layer learns low-level features, such as edges and simple textures. Later intermediate convolutional layers learn middle-level features, such as more complex textures. The deeper convolutional layers learn high-level features, such as objects or parts of objects. Differently, the fully-connected layer, with its larger receptive field, yields global features, which usually are abstract and useful for category-specific discrimination.

The hierarchical structure of CNNs makes it successful in various computer vision tasks. Its feature learning capacity is improved significantly by stacking more convolutional layers, using different filter sizes, or concatenating more convolution operations. Among these DCNNs, there are four models that are widely used as backbone nets for image retrieval.

2.2.2 Popular backbone DCNN architectures

The hierarchical structure and extensive parameterization of DCNNs has led to their success in a remarkable diversity of computer vision tasks. For image retrieval, there are four models which predominantly serve as the networks for feature extraction, including AlexNet [12], VGG [61], GoogLeNet [62], and ResNet [13].

AlexNet is the first DCNN which improved ImageNet classification accuracy by a significant margin compared to conventional methods in ILSVRC 2012. It consists of 5 convolutional layers and 3 fully-connected layers. Input images are usually resized to a fixed size during training and testing stages.

Inspired by AlexNet, VGGNet has two widely used versions: VGG-16 and VGG-19, including 13 convolutional layers and 16 convolutional layers, respectively, but

where all of the convolutional filters are small (local), 3×3 in size. VGGNet is trained in a multi-scale manner where training images are cropped and re-scaled, which improves the feature invariance for the retrieval task.

Compared to AlexNet and VGGNet, GoogLeNet is deeper and wider but has fewer parameters within its 22 layers, leading to higher learning efficiency. GoogLeNet has repeatedly-used inception modules, each of which consists of four branches where 5×5, 3×3, and 1×1 filter sizes are used. These branches are concatenated spatially to obtain the final features for each module. It has been demonstrated that deeper architectures are beneficial for learning higher-level abstract features to mitigate the semantic gap [10].

Finally, ResNet is developed by adding more convolutional layers to extract more abstract features. Skip connections are added between convolutional layers to address the notorious vanishing gradient problem when training this network.

DCNN architectures have developed significantly during the past few years, for which we refer the reader to recent surveys [79, 80]. This chapter focuses on introducing relevant techniques including feature fusion, feature enhancement, and network fine-tuning, based on popular DCNN backbones for performing image retrieval.

2.3 Retrieval with Off-the-Shelf DCNN Models

Because of their size, deep CNNs need to be trained on exceptionally large-scale datasets, and the available datasets of such size are those for image recognition and classification. One possible scheme then, is that deep models effectively trained for recognition and classification directly serve as the off-the-shelf feature detectors for the image retrieval task, the topic of interest in this chapter. That is, one can propose to undertake image retrieval on the basis of DCNNs, trained for classification, and with their pre-trained parameters frozen.

There are limitations with this approach, such that the deep features may not outperform classical hand-crafted features. Most fundamentally, there is a model-transfer or domain-shift issue between tasks [3, 44, 81], meaning that models trained for classification do not necessarily extract features well suited to image retrieval. In particular, a classification decision can be made as long as the features remain within the classification boundaries, therefore the layers from such models may show insufficient capacity for retrieval tasks where feature matching is more important than the final classification probabilities. This section will survey the strategies which have been developed to improve the quality of feature representations, particularly based on feature extraction / fusion (Section 2.3.1) and feature enhancement (Section 2.3.2).

2.3.1 Deep feature extraction

2.3.1.1 Network feedforward scheme

a. Single feedforward pass methods.

Single feedforward pass methods take the whole image and feed it into an off-the-shelf model to extract features. The approach is relatively efficient since the input image is fed only once. For these methods, both the fully-connected layer and last convolutional layer can be used as feature extractors [82].

The fully-connected layer has a global receptive field. After normalization and dimensionality reduction, these features are used for direct similarity measurement without further processing and admitting efficient search strategies [29, 43, 50].

Using the fully-connected layer lacks geometric invariance and spatial information, and thus the last convolutional layer can be examined instead. The research focus associated with the use of convolutional features is to improve their discrimination, where representative strategies are shown in Figure 2.6. For instance, one direction is to treat regions in feature maps as different sub-vectors, thus combinations of different sub-vectors of all feature maps are used to represent the input image.

b. Multiple feedforward pass methods.

Compared to single-pass schemes, multiple pass methods are more time-consuming [3] because several patches are generated from an input image and are both fed into the network before being encoded as a final global feature.

Multiple-pass strategies can lead to higher retrieval accuracy since representations are produced from two stages: patch detection and patch description. Multi-scale image patches are obtained using sliding windows [29, 83] or spatial pyramid model [48], as illustrated in Figure 2.7. However, these patch detection methods lack retrieval efficiency for large-scale datasets since irrelevant patches are also fed into deep networks, thus it is necessary to analyze image patches [31]. As an example, Cao *et al.* [84] propose to merge image patches into larger regions with different hyper-parameters, then the hyper-parameter selection is viewed as an optimization problem under the target of maximizing the similarity between features of the query and the candidates.

Instead of generating multi-scale image patches randomly or densely, region proposal methods introduce a degree of purpose in processing image objects. Region proposals can be generated using object detectors, such as selective search [85] and edge boxes [86]. Aside from using object detectors, region proposals can also be learned using deep networks, such as region proposal networks (RPNs) [17, 54] and convolutional kernel networks (CKNs) [87], and then to apply these deep networks into end-to-end fine-tuning scenarios for learning similarity [88, 89].

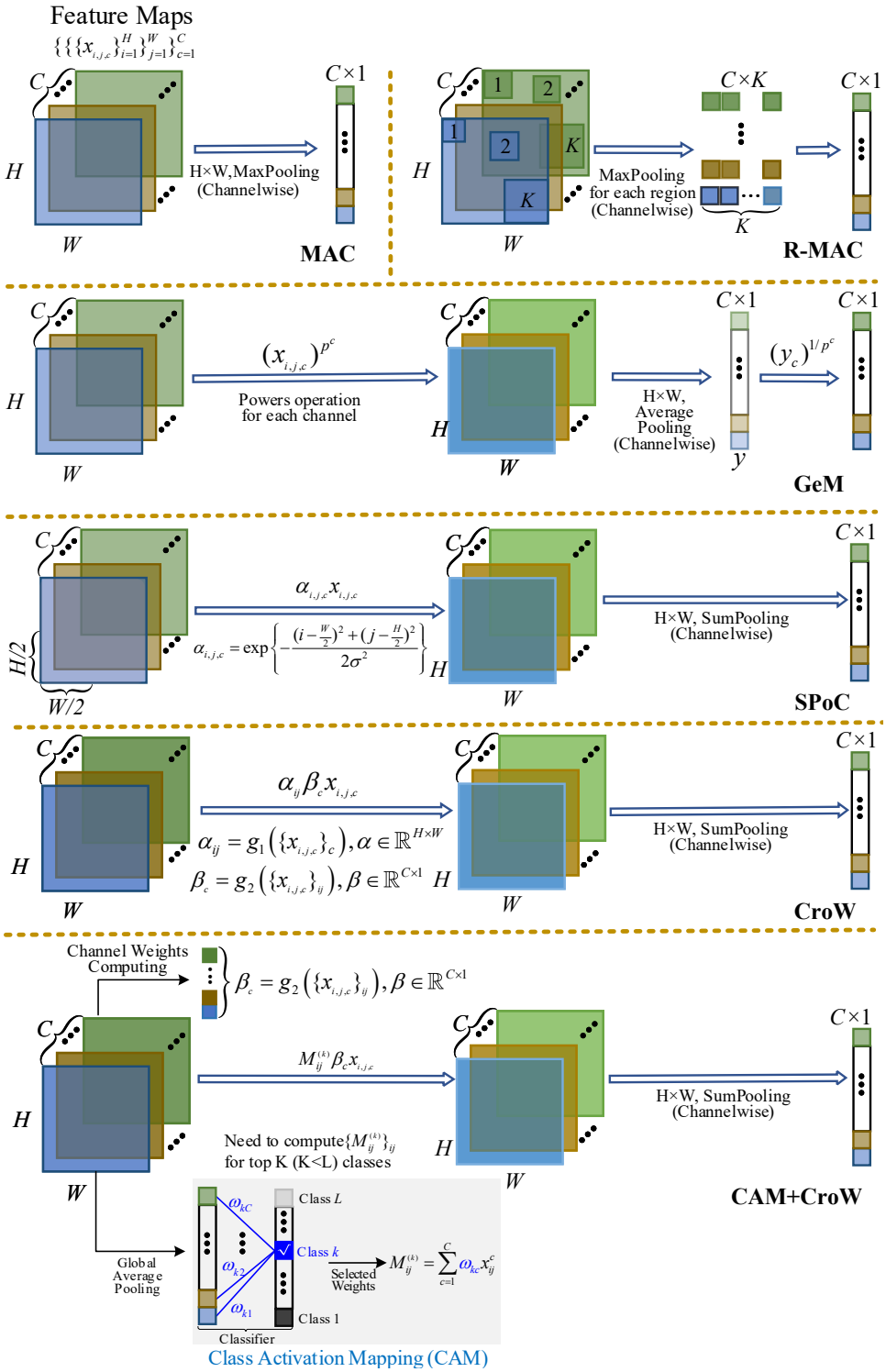


Figure 2.6: Representative methods in single feedforward frameworks, focusing on convolutional feature maps x: MAC [63], R-MAC [31], GeM pooling [57], SPoC with the Gaussian weighting scheme [25], CroW [26], and CAM+CroW [45]. Note that $g_1(\cdot)$ and $g_2(\cdot)$ represent spatial-wise and channel-wise weighting functions, respectively.

2.3.1.2 Deep feature selection

a. Extracted from fully-connected layers

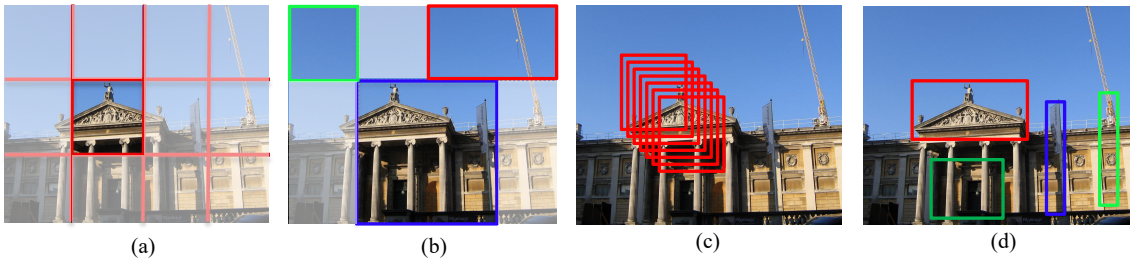


Figure 2.7: Image patch generation schemes: (a) Rigid grid; (b) Spatial pyramid modeling (SPM); (c) Dense patch sampling; (d) Region proposals (RPs) from region proposal networks.

It is straightforward to select a fully-connected layer as a feature extractor [29, 43, 50, 64]. With PCA dimensionality reduction and normalization [43], images' similarity can be measured. Only the fully-connected layer may limit the overall retrieval accuracy, Jun *et al.* [64] concatenate features from multiple fully-connected layers, and Song *et al.* [88] indicate that making a direct connection between the first fully-connected layer and the last layer achieves coarse-to-fine improvements.

As noted, a fully-connected layer has a global receptive field in which each neuron has connections to all neurons of the previous layer. This property leads to two obvious limitations for image retrieval: a lack of spatial information and a lack of local geometric invariance [64].

For the first limitation, researchers focus on the inputs of networks, *i.e.*, using multiple feedforward passes [43]. Compared to taking as input the whole image, discriminative features from the image patches better retain spatial information.

For the second limitation, a lack of local geometric invariance affects the robustness to image transformations such as truncation and occlusion. For this, several works introduce methods to leverage intermediate convolutional layers [25, 29, 63].

b. Extracted from convolutional layers

Features from convolutional layers (usually the last one) preserve more structural details which are especially beneficial for instance-level retrieval [63]. The neurons in a convolutional layer are connected only to a local region of the input feature maps. The smaller receptive field ensures that the produced features preserve more local structural details and are more robust to image transformations like truncation and occlusion [25]. Usually, the robustness of features is improved after pooling.

A convolutional layer arranges the spatial information well and produces location-adaptive features [90]. Various image retrieval methods use convolutional layers as local detectors [25, 30, 31, 45, 63, 90]. For instance, Razavian *et al.* [63] make the first attempt to perform spatial max pooling on the feature maps of an off-the-shelf DCNN model; Babenko *et al.* [25] propose sum-pooling convolutional features

(SPoC) to obtain compact descriptors pre-processed with a Gaussian center prior (see Figure 2.6). Ng *et al.* [90] explore the correlations between activations at different locations on the feature maps, thus improving the final feature descriptor. Yue *et al.* [30] replace BoW [7] with VLAD [28], and are the first to encode local features into VLAD representations. This idea inspired another milestone work [55] where, for the first time, VLAD is used as a layer plugged into the last convolutional layer. The plugged-in layer is end-to-end trainable via back-propagation.

2.3.1.3 Feature fusion strategy

a. Layer-level fusion

Fusing features from different layers aims at combining different feature properties within a feature extractor. It is possible to fuse multiple fully-connected layers in a deep network [64]: For instance, Yu *et al.* [91] explore different methods to fuse the activations from different fully-connected layers and introduce the best-performed P_i-fusion strategy to aggregate the features with different balancing weights, and Jun *et al.* [64] construct multiple fully-connected layers in parallel on the top of ResNet backbone, then concatenate the global features from these layers to obtain the combined global features.

Features from fully-connected layers (global features) and features from convolutional layers (local features) can complement each other when measuring semantic similarity and can, to some extent, guarantee retrieval performance [92].

Global features and local features can be concatenated directly [92, 93]. Before concatenation, convolutional feature maps are filtered by sliding windows or region proposal nets. Pooling-based methods can be applied for feature fusion as well. For example, Li *et al.* [65] propose a Multi-layer Orderless Fusion (MOF) approach, which is inspired by Multi-layer Orderless Pooling (MOP) [29] for image retrieval. However local features can not play a decisive role in distinguishing subtle feature differences because global and local features are treated identically. For this limitation, Yu *et al.* [92] propose using a mapping function to take more advantage of local features in which they are used to refine the return ranking lists. In their work, the exponential mapping function is the key for tapping the complementary strengths of the convolutional layers and fully-connected layers.

It is worth introducing a scheme to explore *which* layer combination is better for fusion given their differences of extracting features. For instance, Chatfield *et al.* [75] demonstrate that fusing convolutional layers and fully-connected layers outperforms the methods that fuse convolutional layers only. In the end, fusing two convolutional layers with one fully-connected layer achieves the best performance.

b. Model-level fusion

It is possible to combine features on different models; such fusion focuses on model complementarity to achieve improved performance, categorized into *intra-model* and *inter-model*.

Generally, intra-model fusion suggests multiple deep models having a similar structure, while inter-model fusion involves models with more differing structures. For instance, Simonyan *et al.* [61] introduce a ConvNet fusion strategy to improve the feature learning capacity of VGG where VGG-16 and VGG-19 are fused. This intra-model fusion strategy reduces the top-5 error by 2.7% in image classification compared to a single counterpart network. Similarly, Ding *et al.* [94] propose a selective deep ensemble framework to combine ResNet-26 and ResNet-50 improve the accuracy of fine-grained instance retrieval. To attend to different parts of the object in an image, Kim *et al.* [95] train an ensemble of three attention modules to learn features with different diversities. Each module is based on different Inception blocks in GoogLeNet.

Inter-model fusion is a way to bridge different features given the fact that different networks have different receptive fields [48, 68, 96, 97, 98]. For instance, a two-stream attention network [68] is introduced to implement image retrieval where the mainstream network for semantic prediction is VGG-16 while the auxiliary stream network for predicting attention maps is DeepFixNet [99]. Considering the importance and necessity of inter-model fusion to bridge the gap between mid-level and high-level features, Liu *et al.* [48] combine VGG-19 and AlexNet to learn combined features, while Ozaki *et al.* [97] make an ensemble to concatenate descriptors from six different models to boost retrieval performance. To illustrate the effect of different parameter choices within the model ensemble, Xuan *et al.* [98] combine ResNet and Inception V1 [62] for retrieval, concentrating on the embedding size and number of embedded features.

Inter-model and intra-model fusion are relevant to model selection. There are some strategies to determine *how* to combine the features from two models. It is straightforward to fuse all types of features from the candidate models and then learning a metric based on the concatenated features [68], which is a kind of "*early fusion*" strategy. Alternatively, it is also possible to learn optimal metrics separately for the features from each model, and then to uniformly combine these metrics for final retrieval ranking [49], which is a kind of "*late fusion*" strategy.

Discussion. Layer-level fusion and model-level fusion are conditioned on the fact that the involved components (layers or whole networks) have different feature description capacities. For these two fusion strategies, the key question is *what features are the best to be combined?* Some explorations have been made for answering this question based on off-the-shelf deep models. For example, Xuan *et al.* [98] illustrate the effect of combining different numbers of features and different sizes within the ensemble. Chen *et al.* [100] analyze the performance of embedded features from image classification and object detection models with respect to image retrieval.

They study the discrimination of feature embeddings of different off-the-shelf models which, to some extent, implicitly guides the model selection when conducting the inter-model level fusion for feature learning.

2.3.2 Deep feature enhancement

2.3.2.1 Feature aggregation

Feature enhancement methods aggregate or embed features to improve the discrimination of deep features. In terms of feature aggregation, sum/average pooling and max pooling are two widely used methods applied on convolutional feature maps. Sum/average pooling is less discriminative, because it considers all activated outputs from a convolutional layer, as a result it weakens the effect of highly activated features [46]. On the contrary, max pooling is particularly well suited for sparse features that have a low probability of being active. Max pooling may be inferior to sum/average pooling if the output feature maps are no longer sparse [101].

Convolutional features can be directly aggregated to produce global ones by spatial pooling. For example, Razavian *et al.* [63, 83] apply max pooling on the convolutional features for retrieval. Babenko *et al.* [25] leverage sum pooling with a Gaussian weighting scheme to encode convolutional features (*i.e.* SPoC). Note that this operation usually is followed by L2 norm and PCA dimensionality reduction.

As an alternative to the holistic approach, it is also possible to pool some regions in a feature map [25, 63], such as done by R-MAC [31]. Also, it is shown that the pooling strategy used in the last convolutional layer usually yields superior accuracy over other shallower convolutional layers and even fully-connected layers.

2.3.2.2 Feature embedding

Apart from direct pooling or regional pooling, it is possible to embed the convolutional features into a high dimensional space to obtain compact ones. The widely used methods include BoW, VLAD, and FV. The embedded features' dimensionality can be reduced using PCA. Note that BoW and VLAD can be extended by using other metrics, such as Hamming distance [102]. Here we briefly describe the principle of the embedding methods for the case of Euclidean distance metric.

BoW [7] is a widely adopted encoding method. BoW encoding leads to a sparse vector of occurrence. Specifically, let $\vec{X} = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}$ be a set of local features, each of which has dimensionality D. BoW requires a pre-defined codebook $\vec{C} = \{\vec{c}_1, \vec{c}_2, ..., \vec{c}_K\}$ with K centroids to cluster these local descriptors, and maps each descriptor $\vec{x}_t$ to the nearest word $\vec{c}_k$. For each centroid $\vec{c}_k$, one can count and normalize the number of occurrences by

$$g(\vec{c}_k) = \frac{1}{T} \sum_{t=1}^{T} \phi(\vec{x}_t, \vec{c}_k) \tag{2.4}$$

$$\phi(\vec{x}_t, \vec{c}_k) = \begin{cases} 1 & \text{if } \vec{c}_k \text{ is the closest codeword for } \vec{x}_t \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Thus BoW considers the number of descriptors belonging to each codebook $\vec{c}_k$ (*i.e.* 0-order feature statistics), then BoW representation is the concatenation of all mapped vectors:

$$G_{_{BoW}}(\vec{X}) = \left[\ g(\vec{c}_1), \cdots, g(\vec{c}_k), \cdots, g(\vec{c}_K)\ \right]^\top \tag{2.6}$$

BoW representation is the histogram of the number of local descriptors assigned to each visual word, so that its dimension is equal to the number of centroids. This method is simple to implement to encode local descriptors, such as convolutional feature maps [65, 82]. However, the embedded vectors are high dimensional and sparse, which are not well suited to large-scale datasets in terms of efficiency.

VLAD [28] stores the sum of residuals for each visual word. Specifically, similar to BoW, it generates K visual word centroids, then each feature $\vec{x}_t$ is assigned to its nearest visual centroid $\vec{c}_k$ and computes the difference $(\vec{x}_t - \vec{c}_k)$:

$$g(\vec{c}_k) = \frac{1}{T}\sum_{t=1}^{T} \phi(\vec{x}_t, \vec{c}_k)(\vec{x}_t - \vec{c}_k) \tag{2.7}$$

where $\phi(\vec{x}_t, \vec{c}_k)$ as defined in (2.5). Finally, the VLAD representation is stacked by the residuals for all centroids, with dimension $(D \times K)$, *i.e.*

$$G_{_{VLAD}}(\vec{X}) = \left[\ \cdots, g(\vec{c}_k)^\top, \cdots\ \right]^\top. \tag{2.8}$$

VLAD captures first order feature statistics, *i.e.* $(\vec{x}_t - \vec{c}_k)$. Similar to BoW, the performance of VLAD is affected by the number of clusters, thereby larger centroids produce larger vectors that are harder to index. For image retrieval, for the first time, Ng *et al.* [30] embed the feature maps from the last convolutional layer into VLAD representations, which is proved to have higher effectiveness than BoW.

The FV method [8] extends BoW by encoding the first and second order statistics continuously. FV clusters the set of local descriptors by a Gaussian Mixture Model (GMM), with K components, to generate a dictionary $C = \{\mu_k; \Sigma_k; w_k\}_{k=1}^{K}$, where w_k, μ_k, Σ_k denote the weight, mean vector, and covariance matrix of the k-th Gaussian component, respectively [103]. The covariance can be simplified by keeping only its diagonal elements, *i.e.*, $\sigma_k = \sqrt{diag(\Sigma_k)}$. For each local feature x_t, a GMM is given by

$$\gamma_k(\vec{x}_t) = w_k \times p_k(\vec{x}_t) / \Big(\sum_{j=1}^{K} w_j p_j(x_t)\Big) \quad s.t. \sum_{j=1}^{K} w_k = 1 \tag{2.9}$$

where $p_k(\vec{x}_t) = \mathcal{N}(\vec{x}_t, \mu_k, \sigma_k^2)$. All local features are assigned into each component k

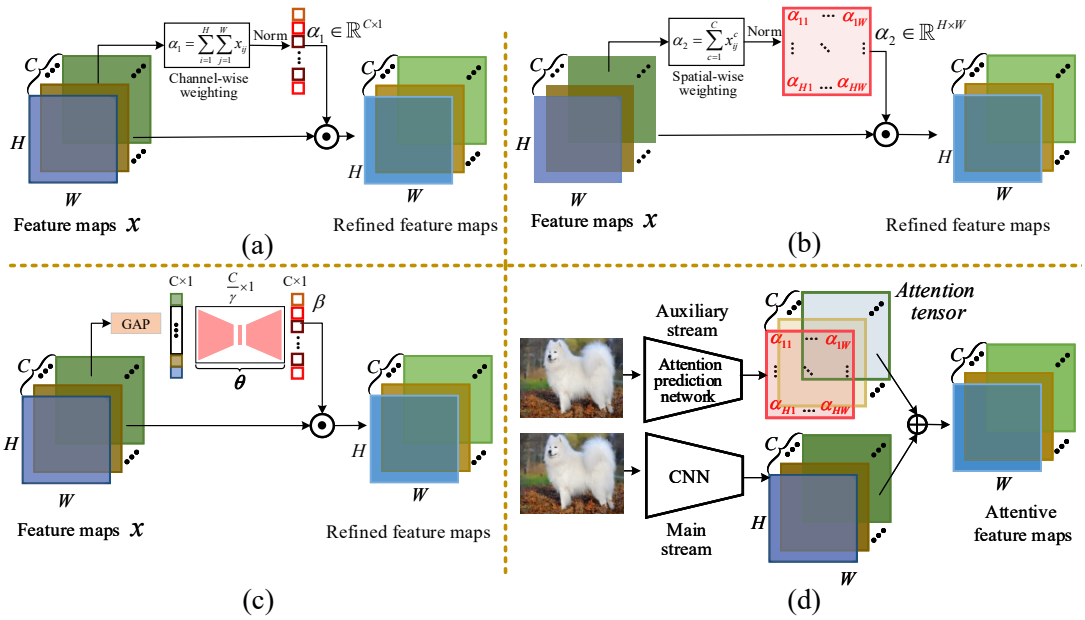


Figure 2.8: Attention mechanisms are shown, divided into two categories. (a)-(b) Non-parametric mechanisms: The attention is based on convolutional feature maps x with size $H \times W \times C$. Channel-wise attention in (a) produces a C-dimensional importance vector α_1 [26, 47]. Spatial-wise attention in (b) computes a 2-dimensional attention map α_2 [26, 45, 74, 90]. (c)-(d) Parametric mechanisms: The attention weights β are provided by a sub-network with trainable parameters (*e.g.* θ in (c)) [105, 106]. Likewise, some off-the-shelf models [99, 107] can predict the attention maps from the input image directly.

in the dictionary, which is computed as

$$
\begin{aligned}
g_{w_k} &= \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} \Big(\gamma_k(\vec{x}_t) - w_k \Big) \\
g_{u_k} &= \frac{\gamma_k(\vec{x}_t)}{T\sqrt{w_k}} \sum_{t=1}^{T} \left(\frac{\vec{x}_t - \mu_k}{\sigma_k} \right), \\
g_{\sigma_k^2} &= \frac{\gamma_k(\vec{x}_t)}{T\sqrt{2w_k}} \sum_{t=1}^{T} \left[\left(\frac{\vec{x}_t - \mu_i}{\sigma_k} \right)^2 - 1 \right]
\end{aligned}
\tag{2.10}
$$

The FV representation is produced by concatenating from the K components:

$$
G_{FV}(\vec{X}) = \left[\; g_{w_1}, \cdots, g_{w_K}, g_{u_1}, \cdots, g_{u_K}, g_{\sigma_1^2}, \cdots, g_{\sigma_K^2} \; \right]^\top \tag{2.11}
$$

The FV representation defines a kernel from a generative process and captures more statistics than BoW and VLAD. FV representations do not increase computational costs significantly but require more memory. Applying FV without memory controls may lead to suboptimal performance [104].

Discussion. Traditionally, sum pooling and max pooling are directly plugged into deep networks and the whole model is used in an end-to-end way, whereas the em-

bedding methods, including BoW, VLAD, and FV, are initially trained separately with pre-defined vocabularies [48, 108]. For these three methods, one needs to pay attention to their properties before choosing one of them to embed deep features. For instance, BoW and VLAD are computed in the rigid Euclidean space where the performance is closely related to the number of centroids. The FV embedding method can capture higher order statistics than BoW or VLAD, thus the FV embedding improves the effectiveness of feature enhancement at the expense of a higher memory cost. Further, when any one of these methods is used, it is necessary to integrate them as a "layer" of deep networks so as to guarantee training and testing efficiency. For example, the VLAD method is integrated into deep networks where each spatial column feature is used to construct clusters via k-means [30]. This idea led to a follow-up approach, NetVLAD [55], where deep networks are fine-tuned with the VLAD vector.

2.3.2.3 Attention mechanisms

The core idea of attention mechanisms is to highlight the most relevant features and to avoid the influence of irrelevant activations, realized by computing an attention map. Approaches to obtain attention maps can be categorized into two groups: non-parametric and parametric-based, as shown in Figure 2.8, where the main difference is whether the importance weights in the attention map are learnable.

Non-parametric weighting is a straightforward method to highlight feature importance. The corresponding attention maps can be obtained by channel-wise or spatial sum-pooling, as in Figure 2.8(a,b). For the spatial-wise pooling of Figure 2.8(b), Kalantidis *et al.* [26] propose a more effective CroW method to weight and pool feature maps. These spatial-wise methods only concentrate on weighting activations at different spatial locations, without considering the relations between these activations. Instead, Ng *et al.* [90] explore the correlations among activations at different spatial locations on the convolutional feature maps. In addition to spatial-wise attention mechanisms, channel-wise weighting methods of Figure 2.8(a) are also popular non-parametric attention mechanisms. Xu *et al.* [47] rank the weighted feature maps to build the "probabilistic proposals" to further select regional features. Jimenez *et al.* [45] combine CroW and R-MAC to propose Classes Activation Maps (CAM) to weight feature maps for each class. Qi *et al.* [66] introduce Truncated Spatial Weighted FV (TSWVF) to enhance the representation of Fisher Vector.

Attention maps can be learned from deep networks, as shown in Figure 2.8(c,d), where the input can be either image patches or feature maps from the previous convolutional layer. The parametric attention methods are more adaptive and are commonly used in supervised metric learning. For example, Li *et al.* [105] propose stacked fully-connected layers to learn an attention model for multi-scale image patches. Similarly, Noh *et al.* [106] design a 2-layer CNN with a softplus output layer to compute scores which indicate the importance of different image regions.

Inspired by R-MAC, Kim *et al.* [109] employ a pre-trained ResNet101 to train a context-aware attention network using multi-scale feature maps.

Instead of using feature maps as inputs, a whole image can be used to learn feature importance, for which specific networks are needed. For example, Mohedano [67] explore different saliency models, including DeepFixNet [99] and Saliency Attentive Model (SAM) [107], to learn salient regions for input images. Similarly, Yang *et al.* [68] introduce a two-stream network for image retrieval in which the auxiliary stream, DeepFixNet, is used specifically for predicting attention maps.

In a nutshell, attention mechanisms offer deep networks the capacity to highlight the most important regions of a given image, widely used in computer vision. For image retrieval specifically, attention mechanisms can be combined with supervised metric learning [90, 95, 110].

2.3.2.4 Deep hash embedding

Real-valued features extracted by deep networks are typically high-dimensional, and therefore are not well-satisfied to retrieval efficiency. As a result, there is significant motivation to transform deep features into more compact codes. Hashing algorithms have been widely used for large-scale image search due to their computational and storage efficiency [77, 111].

Hash functions can be plugged as a layer into deep networks, so that hash codes can be trained and optimized with deep networks simultaneously. During hash function training, the hash codes of originally similar images are embedded as close as possible, and the hash codes of dissimilar images are as separated as possible. A hash function $h(\cdot)$ for binarizing features of an image x may be formulated as

$$b_k = h(x) = h\big(f(x;\boldsymbol{\theta})\big) \quad k = 1, \ldots, K \tag{2.12}$$

then an image can be represented by the generated hash codes $\boldsymbol{b} \in \{+1, -1\}^K$. Because hash codes are non-differentiable their optimization is difficult, so $h(\cdot)$ can be relaxed to be differentiable by using tanh or sigmoid functions [77].

When binarizing real-valued features, it is crucial (1) to preserve image similarity and (2) to improve hash code quality [77]. These two aspects are at the heart of hashing algorithms to maximize retrieval accuracy.

a. Hash functions to preserve image similarity

Preserving similarity seeks to minimize the inconsistencies between the real-valued features and corresponding hash codes, for which a variety of strategies have been adopted.

The design of loss function can significantly influence similarity preservation, which includes both supervised and unsupervised approaches. With the class label avail-

able, many loss functions are designed to learn hash codes in a Hamming space. As a straightforward method, one can optimize the difference between matrices computed from the binary codes and their supervision labels [112]. Other studies regularize hash codes with a center vector, for instance a class-specific center loss is devised to encourage hash codes of images to be close to the corresponding centers, reducing the intra-class variations [111]. Similarly, Kang *et al.* [113] introduce a max-margin t-distribution loss which concentrates more similar data into a Hamming ball centered at the query term, such that a reduced penalization is applied to data points within the ball, a method which improves the robustness of hash codes when the supervision labels may be inaccurate. Moreover metric learning, including Siamese loss [114], triplet loss [51, 115, 116], and adversarial learning [115, 117], is used to retain semantic similarity where only dissimilar pairs keep their distance within a margin. In terms of unsupervised hashing learning, it is essential to capture some relevance among samples, which has been accomplished by using Bayes classifiers [118], KNN graphs [69, 72], k-means algorithms [70], and network structures such as AutoEncoders [119, 120, 121] and generative adversarial networks [60, 69, 122, 123].

Separate from the loss function, it is also important to design deep network frameworks for learning. For instance, Long *et al.* [116] apply unshared-weight CNNs on two datasets where a triplet loss and an adversarial loss are utilized to address the domain shifts. Considering the lack of label information, Cao *et al.* [117] present coined Pair Conditional WGAN, an extension of Wasserstein generative adversarial networks, to generate more samples conditioned on the similarity information.

b. Improving hash function quality

Improving hash function quality aims at making the binary codes uniformly distributed, that is, maximally filling and using the hash code space, normally on the basis of bit uncorrelation and bit balance [77]. Bit uncorrelation implies that different bits are as independent as possible and have little redundancy of information, so that a given set of bits can aggregate more information within a given code length. In principle, bit uncorrelation can be formulated as $\boldsymbol{b}\boldsymbol{b}^\top = \boldsymbol{I}$ in which $\boldsymbol{I}$ is an identity matrix of size K. For example, it can be encouraged via regularization terms such as orthogonality [124] and mutual information [125]. Bit balance means that each bit should have a 50% chance of being +1 or -1, thereby maximizing code variance and information [77]. Mathematically, this condition is constrained by using this regularization term $\boldsymbol{b} \cdot \mathbf{1} = 0$ where $\mathbf{1}$ is a K-dimensional vector with all elements equal to 1.

2.4 Retrieval via Learning DCNN Representations

In Section 2.3, we presented feature fusion and enhancement strategies for which off-the-shelf DCNNs only serve as extractors to obtain features. However, in most cases, deep features may not be sufficient for high accuracy retrieval, even with

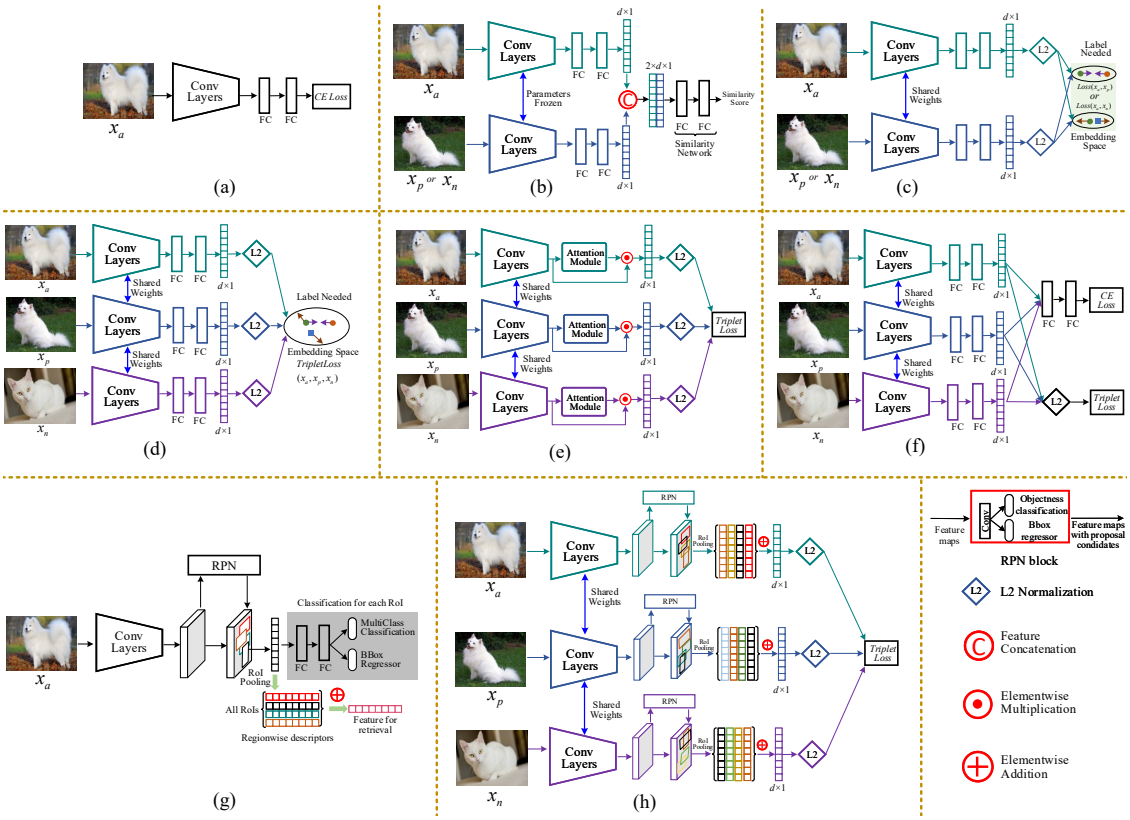


Figure 2.9: Schemes of supervised fine-tuning. Anchor, positive, and negative images are indicated by x_a, x_p, x_n, respectively. (a) classification-based; (b) using a transformation matrix for learning the similarity of image pairs; (c) Siamese networks; (d) triplet loss for fine-tuning; (e) an attention block into DCNNs to highlight regions; (f) combining classification-based and verification-based loss for fine-tuning; (g) region proposal networks (RPNs) to locate the RoI and highlight specific regions or instances; (h) inserting the RPNs of (g) into DCNNs, such that the RPNs extract regions or instances at the convolutional layer.

the strategies which were discussed. In order for models to have higher scalability and to be more effective for retrieval, a common practice is network fine-tuning, *i.e.* updating the pre-stored parameters [44, 78]. However fine-tuning does not contradict or render irrelevant feature processing methods of Section 2.3; indeed, those strategies are complementary and can be incorporated as part of network fine-tuning.

This section focuses on supervised and unsupervised fine-tuning methods for the updating of network parameters.

2.4.1 Supervised fine-tuning

2.4.1.1 Classification-based fine-tuning

When class labels of a new dataset are available, it is preferable to begin with a previously-trained DCNN, trained on a separate dataset, with the backbone DCNN

typically chosen from one of AlexNet, VGG, GoogLeNet, or ResNet. The DCNN can then be subsequently fine-tuned, as shown in Figure 2.9(a), by optimizing its parameters on the basis of a cross entropy loss L_{CE}:

$$L_{CE}(\hat{p}_i, y_i) = -\sum_{i}^{c}\Big(y_i \times log(\hat{p}_i)\Big) \tag{2.13}$$

Here y_i and $\hat{p}_i$ are the ground-truth labels and the predicted logits, respectively, and c is the total number of categories. The milestone work in such fine-tuning is [50], in which AlexNet is re-trained on the Landmarks dataset with 672 pre-defined categories. The fine-tuned network produces superior features on landmark-related datasets like Holidays [126], Oxford-5k, and Oxford-105k [127]. The newly-updated layers are used as global or local feature detectors for image retrieval.

A classification-based fine-tuning method improves the *model-level* adaptability for new datasets, which, to some extent, has mitigated the issue of model transfer for image retrieval. However, there still exists room to improve in terms of classification-based supervised learning. On the one hand, the fine-tuned networks are quite robust to inter-class variability, but may have some difficulties in learning discriminative intra-class variability to distinguish particular objects. On the other hand, class label annotation is time-consuming and labor-intensive for some practical applications. To this end, verification-based fine-tuning methods are combined with classification methods to further improve network capacity.

2.4.1.2 Verification-based fine-tuning

With affinity information indicating similar and dissimilar pairs, verification-based fine-tuning methods learn an optimal metric which minimizes or maximizes the distance of pairs to validate and maintain their similarity. Compared to classification-based learning, verification-based learning focuses on both inter-class and intra-class samples. Verification-based learning involves two types of information [27]:

1. A pair-wise constraint, corresponding to a Siamese network as in Figure 2.9(c), in which input images are paired with either a positive or negative sample;
2. A triplet constraint, associated with triplet networks as in Figure 2.9(e), in which anchor images are paired with both similar and dissimilar samples [27].

These verification-based learning methods are categorized into globally supervised approaches (Figure 2.9(c,d)) and locally supervised approaches (Figure 2.9(g,h)), where the former learn a metric on global features by satisfying all constraints, whereas the latter focus on local areas by only satisfying the given local constraints (*e.g.* region proposals).

To be specific, consider a triplet set X=$\{(x_a, x_p, x_n)\}$ in a mini-batch, where (x_a, x_p) indicates a similar pair and (x_a, x_n) a dissimilar pair. Features $f(x; \boldsymbol{\theta})$ of one image

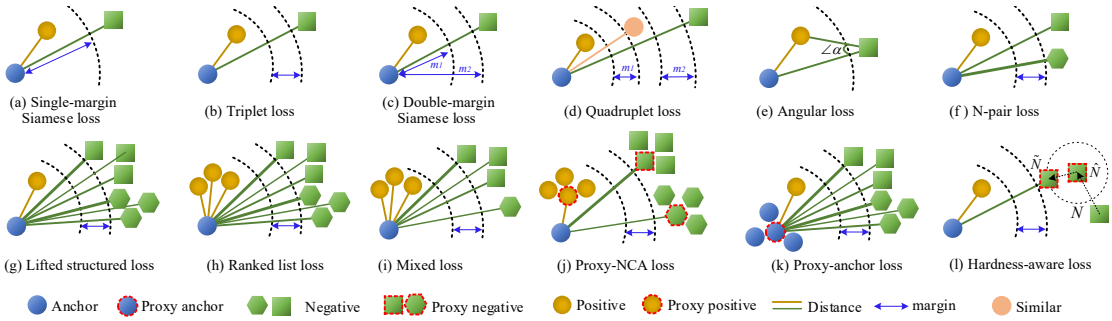


Figure 2.10: Illustrations of sample mining strategies in metric learning. Here, we illustrate three classes, where shapes indicate different classes. Multiple pairs are considered in some loss terms and assigned with distinct weights during training, indicated by different line width. (a)-(c) have been introduced in the text. (d) Quadruplet loss [128]: a sample similar to the anchor is used to construct a double margin. (e) Angular loss [129]: the angle at the negative of triple triangles is computed to obtain higher order geometric constraints. (f) N-pair loss [130]: a positive sample is identified from $N-1$ negative samples of N-1 classes. (g) Lifted structured loss [131]: the structure relationships of three positive and three negative samples are considered. (h) Ranked list loss [132]: all samples to explore intrinsic structured information are considered. (i) Mixed loss [133]: three positive and three negative samples are captured which are initially closely distributed, where another anchor-negative pair initially lies very close to the anchor. (j) Proxy-NCA loss [134]: proxy positive and negative samples for each class are computed and trained with a true anchor sample. (k) Proxy-anchor loss [135]: the anchor sample is represented by a proxy. (l) Hardness-aware loss [136]: the synthetic negative is mapped from an existing hard negative, the hard levels manipulated adaptively within a certain range.

are extracted by a network $f(\cdot)$ with parameters $\boldsymbol{\theta}$, for which we can represent the affinity information for each similar or dissimilar pair as

$$D_{ij} = D(x_i, x_j) = ||f(x_i;\boldsymbol{\theta}) - f(x_j;\boldsymbol{\theta})||_2^2 \tag{2.14}$$

a. Refining with transformation matrix.

Learning the similarity among the input samples can be implemented by optimizing the weights of a linear transformation matrix [52]. It transforms the concatenated feature pairs into a common latent space using a transformation matrix $\boldsymbol{W}\in\mathbb{R}^{2d\times 1}$, where d is the feature dimension. The similarity score of these pairs are predicted via a sub-network $S_W(x_i, x_j) = f_W(f(x_i;\boldsymbol{\theta}) \cup f(x_j;\boldsymbol{\theta}); \boldsymbol{W})$ [52, 137]. In other words, the sub-network f_W predicts how similar the feature pairs are. Given the affinity information of feature pairs $S_{ij} = S(x_i, x_j) \in \{0, 1\}$, the binary labels 0 and 1 indicate the similar (positive) or dissimilar (negative) pairs, respectively. The training of function f_W can be achieved by using a regression loss:

$$L_W(x_i, x_j) = |S_W(x_i, x_j) - S_{ij}\big(sim(x_i, x_j) + m\big) - (1 - S_{ij})\big(sim(x_i, x_j) - m\big)| \tag{2.15}$$

where $sim(x_i, x_j)$ can be the cosine function for guiding training $\boldsymbol{W}$ and m is a

margin. By optimizing the regression loss and updating the transformation matrix $\boldsymbol{W}$, deep networks maximize the similarity of similar pairs and minimize that of dissimilar pairs. It is worth noting that the pre-stored parameters in the deep models are frozen when optimizing $\boldsymbol{W}$. The pipeline of this approach is depicted in Figure 2.9(b) where the weights of the two DCNNs are not necessarily shared.

b. Fine-tuning with Siamese networks.

Siamese networks represent important options in implementing metric learning for fine-tuning, as shown in Figure 2.9(c). It is a structure composed of two branches that share the same weights across the layers. Siamese networks are trained on paired data, consisting of an image pair (x_i, x_j) such that $S(x_i, x_j) \in \{0, 1\}$. A Siamese loss function, illustrated in Figure 2.10(a), is formulated as

$$L_{Siam}(x_i, x_j) = \frac{1}{2}S(x_i, x_j)D(x_i, x_j) + \frac{1}{2}\big(1 - S(x_i, x_j)\big) \max\big(0,\ m - D(x_i, x_j)\big) \tag{2.16}$$

A standard Siamese network and Siamese loss are used to learn the similarity between semantically relevant samples under different scenarios. For example, Simo *et al.* [138] introduce a Siamese network to learn the similarity between paired image patches, which focuses more on the specific regions within an image. Ong *et al.* [53] leverage the Siamese network to learn image features which are then fed into the Fisher Vector model for further encoding. In addition, Siamese networks can also be applied to hashing learning in which the Euclidean distance formulation $D(\cdot)$ in Eq. 2.16 is replaced by the Hamming distance [114].

c. Fine-tuning with triplet networks.

Triplet networks [137] optimize similar and dissimilar pairs simultaneously. As shown in Figure 2.9(d) and Figure 2.10(b), the plain triplet networks adopt a ranking loss for training:

$$L_{Triplet}(x_a, x_p, x_n) = \max\big(0, m + D(x_a, x_p) - D(x_a, x_n))\big) \tag{2.17}$$

which indicates that the distance of an anchor-negative pair $D(x_a, x_n)$ should be larger than that of an anchor-positive pair $D(x_a, x_p)$ by a certain margin m. The triplet loss is used to learn fine-grained image features [71, 96] and for constraining hash code learning [51, 115, 116].

To focus on specific regions or objects, local supervised metric learning has been explored [58, 89, 139, 140]. In these methods, some regions or objects are extracted using region proposal networks (RPNs) [17] which subsequently can be plugged into deep networks and trained in an end-to-end manner, such as shown in Figure 2.9(g), in which Faster R-CNN [17] is fine-tuned for instance search [89]. RPNs yield the regressed bounding box coordinates of objects and are trained by the multi-class classification loss. The final networks extract better regional features by RoI pooling and perform spatial ranking for instance retrieval.

RPNs [17] enable deep models to learn regional features for particular instances or objects [54, 140]. RPNs used in the triplet formulation are shown in Figure 2.9(h). For training, besides the triplet loss, regression loss (PRNs loss) is used to minimize the regressed bounding box according to ground-truth region of interest. In some cases, jointly training an RPN loss and triplet loss leads to unstable results. This is addressed in [54] by first training a CNN to produce R-MAC using a rigid grid, after which the parameters in convolutional layers are fixed and RPNs are trained to replace the rigid grid.

Attention mechanisms can also be combined with metric learning for fine-tuning [110, 139], as in Figure 2.9(e), where the attention module is typically end-to-end trainable and takes as input the convolutional feature maps. For instance, Song *et al.* [139] introduce a convolutional attention layer to explore spatial-semantic information, highlighting regions in images to significantly improve the discrimination for inter-class and intra-class features for image retrieval.

Recent studies [64, 93] have jointly optimized the triplet loss and classification loss function, as shown in Figure 2.9(f). Fine-tuned models that use only a triplet constraint may possess inferior classification accuracy for similar instances [93], since the classification loss does not predict the intra-class similarity, rather locates the relevant images at different levels. Given these considerations, it is natural to combine and optimize triplet constraint and classification loss jointly [64]. The overall joint function is formulated as

$$L_{Joint} = \alpha \cdot L_{Triplet}(x_{i,a}, x_{i,p}, x_{i,n}) + \beta \cdot L_{CE}(\hat{p}_i, y_i) \tag{2.18}$$

where the cross-entropy loss (CE loss) L_{CE} is defined in Eq. (2.13) and the triplet loss $L_{Triplet}$ in Eq. (2.17). α and β are trade-off hyper-parameters to tune the two loss functions.

An implicit drawback of the Siamese loss in Eq. 2.16 is that it may penalize similar image pairs even if the margin between these pairs is small or zero, which may degrade performance [141], since the constraint is too strong and unbalanced. At the same time, it is hard to map the features of similar pairs to the same point when images contain complex contents or scenes. To tackle this limitation, Cao *et al.* [142] adopt a double-margin Siamese loss [141], illustrated in Figure 2.10(c), to relax the penalty for similar pairs. Specifically, the threshold between the similar pairs is set to a margin m_1 instead of being zero. In this case, the original single-margin Siamese loss is re-formulated as

$$\begin{aligned} L(x_i, x_j) = \frac{1}{2} S(x_i, x_j) \max\big(0, D(x_i, x_j) - m_1\big) + \\ \frac{1}{2}\big(1 - S(x_i, x_j)\big) \max\big(0, m_2 - D(x_i, x_j)\big) \end{aligned} \tag{2.19}$$

where $m_1>0$ and $m_2>0$ are the margins affecting the similar and dissimilar pairs, respectively. Therefore, the double margin Siamese loss only applies a contrastive force when the distance of a similar pair is larger than m_1. The mAP metric of retrieval is improved when using the double margin Siamese loss [141].

Discussion. Most verification-based supervised learning methods rely on the basic Siamese or triplet networks. The follow-up studies are focusing on exploring methods to further improve their capacities for robust feature similarity estimation. Generally, the network structure, loss function, and sample selection are important factors for the success of verification-based methods.

A variety of loss functions have been proposed recently [128, 130, 131, 132, 134]. Some of these use more samples or additional constraints. For example, Chen *et al.* [128] incorporate Quadruplet samples for constraining relationships between anchor, positive, negative, and similar images. The N-pair loss [130] and the lifted structured loss [131] even define constraints on all images and employ the structural information of samples in a mini-batch.

The sampling strategy can greatly affect the feature learning and training convergence. To date, many sampling strategies such as clustering have been introduced, of which 12 are shown in Figure 2.10. Aside from sampling within a mini-batch, other work explores mining samples outside a mini-batch even from the whole dataset. This may be beneficial for stabilizing optimization due to a larger data diversity and richer training information. For example, Wang *et al.* [143] propose a cross-batch memory (XBM) mechanism that memorizes the embedding of past iterations, allowing the model to collect sufficient hard negative pairs across multiple mini-batches. Harwood *et al.* [144] provide a framework named smart mining to collect hard samples from the entire training set. It is reasonable to achieve better performance when more samples are used to fine-tune a network. However, the possible additional computational cost during training is a core issue to be addressed.

Directly optimizing the average precision (AP) metric using the listwise AP loss [145] is one way to consider a large number of image simultaneously. Training with this loss has been demonstrated to improve retrieval performance [145, 146, 147], however average precision, as a metric, is normally non-differentiable and non-smooth. To directly optimize the AP loss, the AP metric needs to be relaxed by using methods such as soft-binning approximation [145, 146] or sigmoid function [147].

2.4.2 Unsupervised fine-tuning

Supervised network fine-tuning becomes infeasible when there is not enough supervisory information because such information is costly to assemble or unavailable. Given these limitations, unsupervised fine-tuning methods for image retrieval are quite necessary but less studied [148].

For unsupervised fine-tuning, two broad directions are to mine relevance among features via manifold learning to obtain ranking information, and to devise novel unsupervised frameworks (*e.g.* AutoEncoders), each discussed below.

2.4.2.1 Mining samples with manifold learning

Manifold learning focuses on capturing intrinsic correlations on the manifold structure to mine or deduce revelance, as illustrated in Figure 2.11. Initial similarities between the original extracted features are used to construct an affinity matrix, which is then re-evaluated and updated using manifold learning [149]. According to the manifold similarity in the updated affinity matrix, positive and hard negative samples are selected for metric learning using verification-based loss functions such as pair loss [58, 150], triplet loss [151, 152], or N-pair loss [148], *etc*Note that this is different from the aforementioned methods for verification-based fine-tuning methods, where the hard positive and negative samples are explicitly selected from an ordered dataset according to the given affinity information.

It is important to capture the geometry of the manifold of deep features, generally involving two steps [149] known as a diffusion process. First, the affinity matrix (Figure 2.11) is interpreted as a weighted kNN graph, where each vector is represented by a node, and edges are defined by the pairwise affinities of two connected nodes. Then, the pairwise affinities are re-evaluated in the context of all other elements by diffusing the similarity values through the graph [59, 150, 151, 152]. Some new similarity diffusion methods have recently been proposed, like the regularized diffusion process (RDP) [153] and the regional diffusion mechanism [150]. For more details on diffusion methods we refer to the survey [149].

Most existing algorithms follow a similar principle (*e.g.* random walk [149]). The differences among methods lie primarily in three aspects:

1. **Similarity initialization**, which affects the subsequent KNN graph construction in an affinity matrix. Usually, an inner product [59, 148] or Euclidean distance [56] is directly computed for the affinities. A Guassian kernel function can be used for affinity initialization [149, 152] or Iscen *et al.* [150] consider regional similarity from image patches to build the affinity matrix.

2. **Transition matrix definition**, a row-stochastic matrix [149], determines the probabilities of transiting from one node to another in the graph. These probabilities are proportional to the affinities between nodes, which can be measured by Geodesic distance (*e.g.* the summation of weights of relevant edges).

3. **Iteration scheme**, to re-valuate and update the values in affinity matrix by the manifold similarity until some kind of convergence is achieved. Most algorithms are iteration-based [149, 151], as illustrated in Figure 2.11.

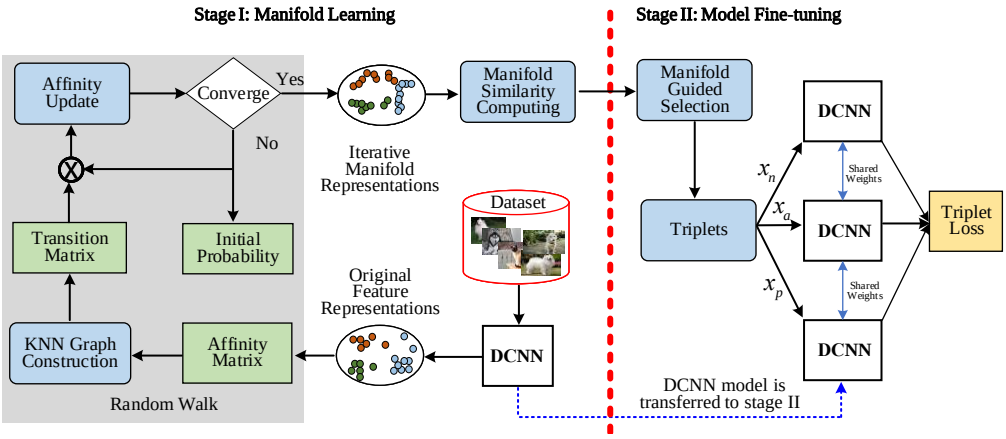


Figure 2.11: Paradigm of manifold learning for unsupervised metric learning, based on triplet loss.

Diffusion process algorithms are indispensable for unsupervised fine-tuning. Better image similarity is guaranteed when it is improved based on initialization (*e.g.* regional similarity [150] or high order information [56]). However, the diffusion process requires more computation and searching due to the iteration scheme [152], a limitation which cannot meet the efficiency requirements of image retrieval. To mitigate this, Nicolas *et al.* [148] apply the closed-form convergence solution of a random walk in each mini-batch to estimate the manifold similarities instead of running many iterations. Some studies replace the diffusion process on a kNN graph with a diffusion network [58], which is derived from graph convolution networks [154]. Their end-to-end framework allows efficient computation during the training and testing stages.

Once the manifold space is learned, samples are mined by computing geodesic distances based on the Floyd-Warshall algorithm or by comparing the set difference [151]. The selected samples are fed into deep networks to perform fine-tuning.

It is possible to explore proximity information, to cluster in Euclidean space, splitting the training set into different groups. For example, Tzelepi *et al.* [155] explore a fully unsupervised fine-tuning method by clustering, in which the kNN algorithm is used to compute the k nearest features, then fine-tuned to minimize the squared distance between each query feature and its k nearest features. As a second example, Radenovic *et al.* [32, 57] use Structure-from-Motion (SfM) for clustering to explore sample reconstructions to select images for triplet loss. Clustering methods depend on the Euclidean distance, making it difficult to reveal the intrinsic relationship between objects.

2.4.2.2 AutoEncoder-based frameworks

An AutoEncoder is a kind of neural network that aims to reconstruct its output as closely as possible to its input. In principle, an input image is encoded as features

into a latent space, and these features are then reconstructed to the original input image using a decoder. The encoder and decoder can be both be convolutional neural networks.

In an AutoEncoder, there exist different levels (*e.g.* pixel-level or instance-level) of reconstruction. These different reconstructions affect the effectiveness of an AutoEncoder, in that pixel-level reconstructions may degrade the learned features of an encoder by focusing on trivial variations in a reconstructed image, since natural images typically contains many detailed factors of location, color, and pose.

An AutoEncoder is an optional framework for supporting other methods, for example the implementation of unsupervised hash learning [60, 119, 120, 121]. Except for the reconstruction loss [60, 121], it is highly necessary to mine feature relevance to explore other objective functions. This is usually realized by using clustering algorithms [121] since features from an off-the-shelf network initially contain rich semantic information to keep their semantic structure [69, 72, 118]. For example, Gu *et al.* [121] introduce a modified cross-entropy based on the k-means clustering algorithm where a deep model learns to cluster iteratively and yields binary codes while retaining the structures of the input data distributions. Zhou *et al.* [72] and Deng *et al.* [69] propose a self-taught hashing algorithm using a kNN graph construction to generate pseudo labels that are used to analyze and guide network training. Other techniques such as Bayes Nets are also used to predict sample similarity, such as in the work of Yang *et al.* [118], which adopts a Bayes optimal classifier to assign semantic similarity labels to data pairs which have a higher similarity probability.

AutoEncoders can also be integrated into other frameworks, such as graph convolutional networks [154] and object detection models [156] to learn better binary latent variables. For example, Shen *et al.* [60] combine graph convolutional networks [154] to learn the hash codes from an AutoEncoder. In this method, the similarity matrix for graph learning is computed on the binary latent variables from the Encoder. Generative adversarial networks (GANs) are also explored in the unsupervised hashing framework [60, 69, 122, 123]. The adversarial loss in GANs is the classical objective to use. By optimizing this loss, the synthesized images generated from hash codes gradually keep semantic similarity consistent for the original images. The pixel-level and feature-level content loss are used to improve the generated image quality [122]. Some other losses are employed in GANs to enhance hash code learning. For instance, a distance matching regularizer is utilized to propagate the correlations between high-dimensional real-valued features and low-dimensional hash codes [157], or two loss functions that aim at promoting independence of binary codes [123]. In summary, using GANs for unsupervised hash learning is promising, but there remains much room for further exploration.

2.5 State of the Art Performance

2.5.1 Datasets

To demonstrate the effectiveness of methods, we choose four commonly-used datasets for performance comparison: Holidays, Oxford-5k (including the extended Oxford-105k), Paris-6k (including the extended Paris-106k) and UKBench.

UKBench (UKB) [158] consists of 10,200 images of objects. The whole dataset has 2,550 groups of images, each group having four images of the same object from different viewpoints or illumination conditions. Each image in the dataset can be used as a query image.

Holidays [126] consists of 1,491 images collected from personal holiday albums. Most images are scene-related. The dataset comprises 500 groups of similar images with a query image for each group. In each group, the first image is used as a query image for performance evaluation.

Oxford-5k [127] consists of 5,062 images for 11 Oxford buildings. Each image is represented by five queries by a hand-drawn bounding box, thus there are 55 query Regions of Interest (RoI) in total. An additional disjoint set of 100,000 distractor images is added to obtain Oxford-100k.

Paris-6k [159] includes 6,412 images collected from Flickr. It is categorized into 12 groups about specific Paris architectures. The dataset has 500 query images for evaluation, and 55 queries with bounding boxes. Images are annotated with the same four types of labels as used in the Oxford-5k dataset.

Annotations and evaluation protocols in Oxford-5k and Paris-6k are updated; additional queries and distractor images are added into the two datasets, producing the *Revisited Oxford* and *Revisited Paris* datasets [160]. Due to the popularity of Oxford-5k and Paris-6k, we primarily undertake performance evaluations on the original datasets.

2.5.2 Evaluation metrics

Average precision (AP) refers to the coverage area under the precision-recall curve. A larger AP implies a higher precision-recall curve and better retrieval accuracy. AP can be calculated as

$$AP = \frac{\sum_{k=1}^{N} P(k) \cdot rel(k)}{R} \tag{2.20}$$

where R denotes the number of relevant results for the query image from the total number N of images. $P(k)$ is the precision of the top k retrieved images, and $rel(k)$ is an indicator function equal to 1 if the item within rank k is a relevant image and 0 otherwise. Mean average precision (mAP) is adopted for the evaluation over all

query images,

$$\frac{1}{Q}\sum_{q=1}^{Q} AP(q) \tag{2.21}$$

where Q is the number of query images.

Additionally, N-S score is a metric used for UKBench [158]. In this dataset, there are four relevant images for each query. The N-S score is the average, four times, for the top-four precision over the dataset.

2.5.3 Performance comparison and analysis

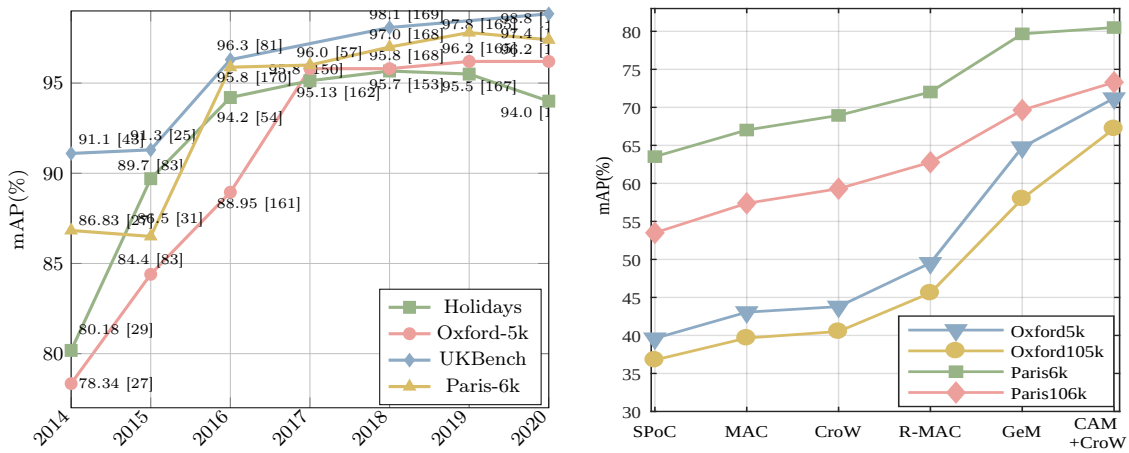
Overview. We conclude with the performance over these 4 datasets from 2014 to 2020 in Figure 2.12(a). At early period, DCNNs acted as powerful extractors and achieved good results, *e.g.* mAP is 78.34% in [27] on Oxford-5k. Subsequently, the results increased significantly when some crucial factors were adopted, including feature fusion [161, 162, 163], feature aggregation [31, 63], and network fine-tuning [153, 164]. For instance, the accuracy on UKBench reaches an mAP of 98.8% in [163] when an undirected graph is defined to fuse features and estimate their correlations. Network fine-tuning improves performance greatly. The accuracy increases steadily from 78.34% [27] to 96.2% [165] on the Oxford-5k dataset when manifold learning is used to fine-tune deep networks.

We evaluate the methods using off-the-shelf models (Table 2.2) and fine-tuning networks (Table 2.3). In Table 2.2, single pass and multiple pass are analyzed, while supervised fine-tuning and unsupervised fine-tuning are compared in Table 2.3.

Evaluation for single feedforward pass. The common practice using this scheme is to enhance feature discrimination. In Table 2.2, we observe that fully-connected layers used as feature extractors may reach a lower accuracy (*e.g.* 74.7% on Holidays in [50]), compared to the counterpart convolutional layers because the fully-connected layers lack structural information. Layer-level feature fusion strategy improves retrieval accuracy. For example, Yu *et al.* [92] combined three layers (*Conv4*, *Conv5*, and *FC6*) (*e.g.* an mAP of 91.4% on Holidays), outperforming the performance of non-fusion method in [25] (*e.g.* mAP is 80.2%). Moreover, convolutional features embedded by BoW model reach a competitive performance on Oxford-5k and Paris-6k (73.9% and 82.0%, respectively), while its codebook size is 25k, which may affect the retrieval efficiency. For single pass scheme, methods shown in Figure 2.6 improve the discrimination of convolutional feature maps and perform differently in Table 2.2 (*e.g.* 66.9% of R-MAC [159], 58.9% of SPoC [25] on Oxford-5k). We view this as a critical factors and further analyze.

Evaluation for multiple feedforward pass. The methods exemplified in Figure 2.7 are reported their results in multiple pass scheme. Among them, extracting image patches densely using Overfeat [166] can reach best results on the 4 datasets [43].

Using rigid grid method reach competitive results (*e.g.* an mAP of 87.2% on Paris-6k) [108]. These two methods consider more patches, even background information when used for feature extraction. Instead of generating patches densely, region proposals and spatial pyramid modeling have a degree of purpose in processing image objects. This may be more efficient and less memory demanding. Using multiple-pass scheme, spatial information is maintained better than the case using the single-pass method. For example, a shallower network (AlexNet) and region proposal networks are used in [85], its result on UKBench is 3.81 (N-Score), higher than the one using deeper networks, such as [25, 50, 92]. Besides feeding image patches into the same network, model-level fusion also exploit complementary spatial information to improve the retrieval accuracy. For instance, as reported in [48], which combines AlexNet and VGG, the results on Holidays (81.74% of mAP) and UKBench (3.32 of N-Score) are better than these in [65] (76.75% and 3.00, respectively).



(a) (b)

Figure 2.12: (a) Performance improvement from 2014 to 2020. (b) mAP comparison of the feature aggregation methods shown in Figure 2.6.

Evaluation for supervised fine-tuning. Compared to the off-the-shelf models, fine-tuning deep networks usually improves accuracy, see Table 2.3. For instance, the result on Oxford-5k [31] by using a pre-trained VGG is improved from 66.9% to 81.5% in [53] when a single-margin Siamese loss is used. Similar trends can be also observed on the Paris-6k dataset. Although classification-based fine-tuning method is not excel at learning intra-class variability (*e.g.* an mAP of 55.7% on Oxford-5k in [50]), its performance may be improved with powerful DCNNs and feature enhancement methods such as the attention mechanism in [106], leading to an mAP of 83.8% on Oxford-5k. As for verification-based fine-tuning methods, in some cases, the loss used for fine-tuning is essential for performance improvement. For example, RPN is re-trained using regression loss on Oxford-5k and Paris-6k (75.1% and 80.7%, respectively) [89]. Its results are lower than the results from [52] (88.2% and 88.2%, respectively) where a transformation matrix is used to learn

Table 2.1: Evaluations of mAP (%), N-S score, and average search time per image. "†" refers to the query time is evaluated in a global diffusion manner, while "‡" refers to the time is evaluated in a regional diffusion way.

	Oxford-5k (+100k)		Paris-6k (+100k)		Holidays		UKB	
	mAP	Time	mAP	Time	mAP	Time	N-S	Time
[153]	91.3 (88.4)	5.45 *ms* (809 *ms*)	-	-	95.66	3.11 *ms*	3.93	4.91 *ms*
[165]	92.6 (91.8)	2 *ms* (10 *ms*)	-	-	-	-	-	-
[150]†	85.7 (-)	20 *ms* (-)	94.1 (-)	20 *ms* (-)	-	-	-	-
[150]‡	95.8 (-)	600 *ms* (-)	96.9 (-)	700 *ms* (-)	-	-	-	-
[172]	64.9 (58.8)	0.81 *ms* (0.82 *ms*)	-	-	-	-	-	-
[57]	64.8 (57.9)	0.77 *ms* (0.73 *ms*)	-	-	-	-	-	-
[52]	55.5 (-)	0.35 *ms* (-)	71.0 (-)	0.35 *ms* (-)	-	-	-	-

visual similarity. However, when RPN is trained by using triplet loss such as [140], the effectiveness of retrieval is improved significantly where the results are 86.1% (on Oxford-5k) and 94.5% (on Paris-6k). Further, feature embedding methods are important for retrieval accuracy. For example, Ong *et al.* [53] embedded *Conv5* feature maps by Fisher Vector and achieved an mAP of 81.5% on Oxford-5k, while embedding feature maps by using VLAD achieves an mAP of 62.5% on this dataset [32, 55].

Evaluation for unsupervised fine-tuning. Compared to supervised fine-tuning, unsupervised fine-tuning methods are relatively less explored. The difficulty for unsupervised fine-tuning is to mine relevance of samples without ground-truth labels. In general, unsupervised fine-tuning methods produce lower performance than the supervised fine-tuning methods. For instance, supervised fine-tuning network by using Siamese loss in [171] achieves an mAP 88.4% on Holidays, while unsupervised fine-tuning network using the same loss function in [32, 57, 151] achieve 82.5%, 83.1%, and 87.5%, respectively. However, unsupervised fine-tuning methods can achieve a similar accuracy even outperform the supervised fine-tuning if a suited feature embedding method is used. For instance, Zhao *et al.* [152] explore global feature structure with modeling the manifold learning, producing an mAP of 85.4% (on Oxford-5k) and 96.3% (on Paris-6k). This is similar to the supervised method [140], whose results are 86.1% (on Oxford-5k) and 94.5% (on Paris-6k). As another example, the precision of ResNet-101 fine-tuned by cross-entropy loss achieves to 83.8% on Oxford-5k [106], while the precision is further improved to 92.0% when IME layer is used to embed features and fine-tuned in an unsupervised way [56]. Note that fine-tuning strategies are related to the type of the target retrieval datasets.

Retrieval efficiency is also an important criterion for image retrieval. Deep learning methods are usually trained and validated on large-size datasets, relying on using

GPUs. Most prior works focus more on retrieval accuracy but less on efficiency. We report the retrieval accuracy and retrieval efficiency on the 4 datasets in Table 2.1. The recorded time (in ms) indicates the average time for searching each query image. In Table 2.1, we observe some important trends. First, in general, the average retrieval time for each query image is less than $1s$. Concretely, the recorded time is up to $809ms$ on Oxford-105k in [153], whose mAP is 88.4%. The retrieval time is $600ms$ on Oxford-5k and $700ms$ on Paris-6k in [150], whose time cost is caused by processing 21 regional features on each query image. Second, we observe the retrieval accuracy-efficiency balancing issue, which is significantly obvious on the Oxford-5k dataset. The average retrieval time are both less than $1ms$ in prior work [52, 57, 172], whose mAPs are lower than 70% (*i.e.* 55.5%, 64.8%, and 64.9%, respectively). In contrast, the prior approaches [150, 153, 165], reach relatively higher mAPs (*i.e.* 91.3%, 92.6%, and 95.8%, respectively), while this higher accuracy is at the expense of efficiency (more than $2ms$ even up to $600ms$). Therefore, the trade-off of accuracy and efficiency is also an important factor to take into account in deep image retrieval, especially for large-scale datasets.

In addition, we discuss other important factors, including the depth of networks, retrieval feature dimension, and feature aggregation methods.

Network depth. We compare the efficacy of DCNNs depth, following the fine-tuning protocols[1] in [57]. For fair comparisons, all convolutional features from these backbone DCNNs are aggregated by MAC method [63], and fine-tuned by using the same learning rate. That means, the adopted methods are the same except the DCNNs have different depths. We use the default feature dimension (*i.e.* AlexNet (256-d), VGG (512-d), GoogLeNet (1024-d), ResNet-50/101 (2048-d)). The results are reported in Figure 2.13(a). We observe that the deeper networks is more beneficial for accuracy boosts, due to extracting more discriminative features.

Feature dimension. We vary the feature dimension of ResNet-50 from 32-d to 8192-d, by adding fully-connected layers on the top of pooled convolutional features. The results are shown in Figure 2.13(b). It is expected that higher-dimensional features capture much more semantics and are beneficial for retrieval. However, the performance tends to be stable when the dimension is very large. For ResNet-50, we observe that the 2048-d feature can already produce competitive results.

Feature aggregation methods. Here, we further discuss the methods of embedding convolutional feature maps, as illustrated in Figure 2.6. We use the off-the-shelf VGG (without updating parameters) on the Oxford and Paris datasets. The results are reported in Figure 2.12(b). We observe that different ways to aggregate the same off-the-shelf DCNN make differences for retrieval performance. These reported results provide a reference for feature aggregation when one uses convolutional layers for performing retrieval tasks.

[1]https://github.com/filipradenovic/cnnimageretrieval-pytorch

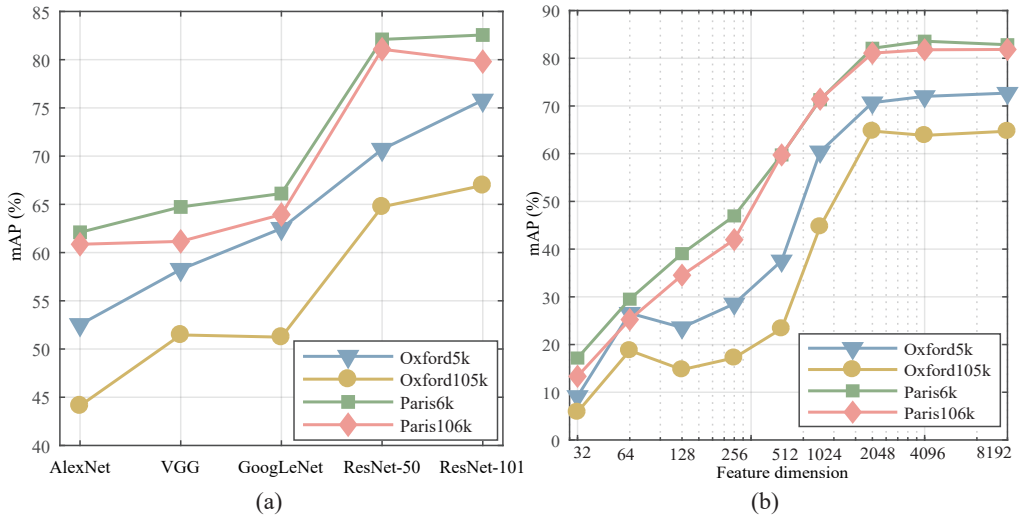


Figure 2.13: (a) The effectiveness of different DCNNs on 4 datasets. All models are fine-tuned by the same loss function. The results are tested on the convolutional features with default dimension; (b) The impact of feature dimension on retrieval performance. These features are extracted by using ResNet-50.

2.6 Chapter Conclusions

In this chapter, we reviewed deep learning methods for image retrieval, and categorized it into deep image retrieval of off-the-shelf models and fine-tuned models according to the parameter updates of deep networks. Concretely, the off-the-shelf group is concerned with obtaining high-quality features by freezing the pre-stored parameters where network feedforward schemes, layer selection, and feature fusion methods are presented. While fine-tuned based methods deal with updating networks with optimal parameters for feature learning in both supervised and unsupervised approaches. For each group, we presented the corresponding methods and compared their differences. The corresponding experimental results are collected and analyzed for all the categorized works.

Deep learning has shown significant progress and spotlighted its capacity for image retrieval. Despite the great success, there are still many unsolved problems. Here, we introduce some promising trends as future research directions. We hope that this chapter not only provides a better understanding of image retrieval but also facilitates future research activities and application developments in this field.

Table 2.2: Performance evaluation of off-the-shelf DCNN models. "•" indicates that the models or layers are combined to learn features; "PCA_w indicates PCA with whitening on the extracted features to improve robustness; "MP" means Max Pooling; "SP" means Sum Pooling. The CNN-M network with "$*$" has an architecture similar to that of AlexNet. "-" means that the results were not reported.

Type	Method	Backbone DCNN	Output Layer	Feature Enhance.	Feature Dimension	Holidays	UKB	Oxford5k (+100k)	Paris6k (+100k)	Brief Conclusions and Highlights
Single Pass	Neural codes [50]	AlexNet	FC6	PCA	128	74.7	3.42 (N-S)	43.3 (38.6)	-	Compressed neural codes of different layers are explored. AlexNet is also fine-tuned for retrieval.
	R-MAC [31]	VGG16	Conv5	R-MAC + PCA_w	512	-	-	66.9 (61.6)	83.0 (75.7)	Adopting sliding windows with different scales on the convolutional feature maps to preserve spatial information.
	CroW [26]	VGG16	Conv5	CroW + PCA_w	256	85.1	-	68.4 (63.7)	76.5 (69.1)	The spatial- and channel-wise weighting mechanisms are utilized to highlight crucial convolutional features.
	BLCF [82]	VGG16	Conv5	BoW + PCA_w	25k	-	-	73.9 (59.3)	82.0 (64.8)	Both global features and local features are explored, demonstrating that local features have higher accuracy.
	SPoC [25]	VGG16	Conv5	SPoC + PCA_w	256	80.2	3.65 (N-S)	58.9 (57.8)	-	Exploring Gassian weighting scheme *i.e.* the centering prior, to improve the discrimination of features.
	Multi-layer CNN [92]	VGG16	FC6 • Conv4∼5	SP	4096	91.4	3.68 (N-S)	61.5 (-)	-	Layer-level feature fusion and the complementary properties of different layers are explored.
Multiple Pass	Deepindex [48]	AlexNet • VGG19	FC6-7 • FC17-18	BoW + PCA	512	81.7	3.32 (N-S)	-	75.4 (-)	Exploring layer-level and model-level fusion methods. Image patches are extracted using spatial pyramid modeling.
	MOF [65]	CNN-M* [75]	FC7 • Conv	SP or MP + BoW	20k	76.8	3.00 (N-S)	-	-	Exploring layer-level fusion scheme. Image patches are extracted using spatial pyramid modeling.
	Multi-scale CNN [63]	VGG16	Conv5	SP or MP + PCA_w	32k	89.6	95.1 (mAP)	84.3 (-)	87.9 (-)	Image patches are extracted in a dense manner. Geometric invariance is considered when aggregating patch features.
	CNNaug-ss [43]	Overfeat [166]	FC	PCA_w	15k	84.3	91.1 (mAP)	68.0 (-)	79.5 (-)	Image patches are extracted densely. Image regions at different locations with different sizes are included.
	MOP-CNN [29]	AlexNet	FC7	VLAD + PCA_w	2048	80.2	-	-	-	Image patches are extracted densely. Multi-scale patch features are further embedded into VLAD descriptors.
	CCS [73]	GoogLeNet	Conv	VLAD + PCA_w	128	84.1	3.81 (N-S)	64.8 (-)	76.8 (-)	Object proposals are extracted by RPNs. Object-level and point-level feature concatenation schemes are explored.
	OLDFP [85]	AlexNet	FC6	MP + PCA_w	512	88.5	3.81 (N-S)	60.7 (-)	66.2 (-)	Exploring the impact of proposal number. Patches are extracted by RPNs and the features are encoded in an orderless way.
	LDD [108]	VGG19	Conv5	BoW + PCA_w	500k	84.6	-	83.3 (-)	87.2 (-)	Image patches are obtained using a uniform square mesh. Patch features are encoded into BoW descriptors.

Table 2.3: Performance evaluation of methods in which DCNN models are fine-tuned, in a supervised or an unsupervised manner. "CE Loss" means the models are fine-tuned using the classification-based loss function in the form of Eq. 2.13. "Siamese Loss" is in the form of Eq. 2.16. "Regression Loss" is in the form of Eq. 2.15. "Triplet Loss" is in the form of Eq. 2.17.

Type	Method	Backbone DCNN	Output Layer	Feature Enhance.	Loss Function	Feature Dimension	Holidays	UKB	Oxford5k (+100k)	Paris6k (+100k)	Brief Conclusions and Highlights
Supervised Fine-tuning	DELF [106]	ResNet-101	Conv4 Block	Attention + PCA_w	CE Loss	2048	-	-	83.8 (82.6)	85.0 (81.7)	Exploring the FCN to construct feature pyramids of different sizes.
	Neural codes [50]	AlexNet	FC6	PCA	CE Loss	128	78.9	3.29 (N-S)	55.7 (52.3)	-	The first work which fine-tunes deep networks for image retrieval. Compressed neural codes and different layers are explored.
	Non-metric [52]	VGG16	Conv5	PCA_w	Regression Loss	512	-	-	88.2 (82.1)	88.2 (82.9)	Visual similarity learning of similar and dissimilar pairs is performed by a neural network, optimized using regression loss.
	Faster R-CNN [89]	VGG16	Conv5	MP / SP	Regression Loss	512	-	-	75.1 (-)	80.7 (-)	RPN is fine-tuned, based on bounding box coordinates and class scores for specific region query which is region-targeted.
	SIAM-FV [53]	VGG16	Conv5	FV + PCA_w	Siamese Loss	512	-	-	81.5 (76.6)	82.4 (-)	Fisher Vector is integrated on top of VGG and is trained with VGG simultaneously.
	SIFT-CNN [171]	VGG16	Conv5	SP	Siamese Loss	512	88.4	3.91 (N-S)	-	-	SIFT features are used as supervisory information for mining positive and negative samples.
	Quartet-Net [142]	VGG16	FC6	PCA	Siamese Loss	128	71.2	87.5 (mAP)	48.5 (-)	48.8 (-)	Quartet-net learning is explored to improve feature discrimination where double-margin contrastive loss is used.
	NetVLAD [55]	VGG16	VLAD Layer	PCA_w	Triplet Loss	256	79.9	-	62.5 (-)	72.0 (-)	VLAD is integrated at the last convolutional layer of VGG16 network as a plugged layer.
	Deep Retri-eval [140]	ResNet-101	Conv5 Block	MP + PCA_w	Triplet Loss	2048	90.3	-	86.1 (82.8)	94.5 (90.6)	Dataset is cleaned automatically. Features are encoded by R-MAC. RPN is used to extract the most relevant regions.
Unsupervised Fine-tuning	MoM [151]	VGG16	Conv5	MP + PCA_w	Siamese Loss	64	87.5	-	78.2 (72.6)	85.1 (78.0)	Exploring manifold learning for mining dis/similar samples. Features are tested globally and regionally.
	GeM [57]	VGG16	Conv5	GeM Pooling	Siamese Loss	512	83.1	-	82.0 (76.9)	79.7 (72.6)	Fine-tuning CNNs on an unordered dataset. Samples are selected from an automated 3D reconstruction system.
	SfM-CNN [32]	VGG16	Conv5	PCA_w	Siamese Loss	512	82.5	-	77.0 (69.2)	83.8 (76.4)	Employing Structure-from-Motion to select positive and negative samples from unordered images.
	IME-CNN [56]	ResNet-101	IME Layer	MP	Regression Loss	2048	-	-	92.0 (87.2)	96.6 (93.3)	Graph-based manifold learning is explored within an IME layer to mine the matching and non-matching pairs in unordered datasets.
	MDP-CNN [152]	ResNet-101	Conv5 Block	SP	Triplet Loss	2048	-	-	85.4 (85.1)	96.3 (94.7)	Exploring global feature structure by modeling the manifold learning to select positive and negative pairs.

Chapter 3

Domain Uncertainty based on Information Theory for Cross-modal Hash Retrieval

In the previous chapter, we gave a comprehensive review about intelligent image retrieval. Semantic information that helps us understand the world usually comes from different modalities. We can express the same concept by using different ways so that we can search the images of interest by submitting any media content at hand (*e.g.* a phrase, or an image) as the query item. Therefore, cross-modal hash retrieval, as a natural searching way, has received considerable interest in the area of deep learning. Here hash codes of data of different modalities are learned where pair-wise loss functions control feature similarity in a shared embedding space. In this chapter, we improve on feature similarity by using Shannon's information entropy with respect to the modality information that is present in learning superior hash codes. We introduce a novel network for predicting the domain from the learned features while the protagonist network uses a loss function based on Shannon's information entropy to learn to maximize the domain uncertainty and therefore the information content. Additionally, according to the number of common labels between each similar image-text pair, we define a multi-level similarity matrix as supervisory information, which constrains all similar pairs with different weights. We show with extensive experiments that our novel approach to domain uncertainty leads to a cross-modal hash retrieval that outperforms the state-of-the-art.

Keywords

Information entropy, cross-modal hash retrieval, domain uncertainty, multi-level similarity

This chapter is based on the following publication [34]:

- Chen, W., Pu, N., Liu, Y., Bakker, E. and Lew, M.S., "Domain Uncertainty Based On Information Theory for Cross-Modal Hash Retrieval." IEEE International Conference on Multimedia and Expo (ICME), 2019, pp 43-48.

3.1 Introduction

Cross-modal retrieval has been a compelling research topic in recent years [173, 174, 175]. It aims to accurately index semantically relevant samples from one modality, such as finding a text that describes a given image and vice versa. Meanwhile, to optimize retrieval and storage costs, binary representation learning (*a.k.a* hash code learning) has received increasing attention. Reducing the heterogeneity gap [176] and the semantic gap [10] (*i.e.* retaining feature similarity) are two key issues being explored in cross-modal hash retrieval. Since the data in different modalities are described by different statistical properties, the heterogeneity gap characterizes the difference between feature vectors from different modalities that have similar semantics but are distributed in different spaces. Similarities between these feature vectors are not well associated so that these vectors are not directly comparable, leading to inconsistent distributions. The semantic gap characterizes the difference, in any application, between the high-level concepts of humans and the low-level features typically derived from images (*i.e.* pixels or symbols) [10].

Convolutional Neural Networks (CNNs) have demonstrated powerful feature learning capacity. Discriminative features for each modality are separately learned well using deep learning methods. However, features from different modalities have usually heterogeneous distributions and representations. Textual features are often more abstract than visual features. A common practice is to map features for different modalities into a common Hamming space where hash codes can be assessed directly and the heterogeneity gap is diminished. Existing methods for feature projection are categorized into unsupervised [174] and supervised [173, 175]. Compared to unsupervised methods, supervised hash approaches can achieve superior performance with the help of semantic labels or relevant information.

In recent years, metric learning is used to retain feature similarity when projecting modality features into a common space, such as ranking loss [177], and contrastive loss [173, 178]. In the common space, features of similar pairs are projected together, while for dissimilar pairs features will be pushed away. These loss functions focus on each pair separately and learn their features according to their affinity information. However, using these loss functions cannot guarantee that the feature distributions for image and text are consistent. To tackle this limitation, adversarial learning is incorporated to study the levels of agreement between feature distributions from image and text when classified into their corresponding modality labels [175, 177, 178]. To obtain a suitable common space, the gradients need to be reversed by the optimizing adversarial networks. However, there still exist some limitations. First, discrimination for image and text will tend to the semantically-similar image-text pairs far away because they belong to different modalities; Second, modality labels are needed in adversarial learning which limits the generalization to these cases where modalities are not just image and text; Third, the gradient reversal in adversarial learning is not straightforward.

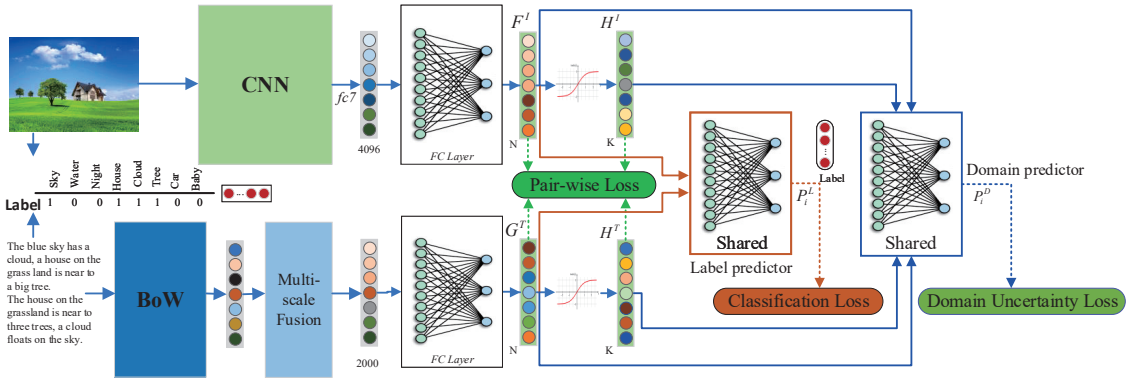


Figure 3.1: The framework for cross-modal hash retrieval. Domain uncertainty loss is based on information theory (Section 3.3.1); Pair-wise loss is constrained by binary similarity matrix **S** and multi-level similarity matrix **W** (Section 3.3.1); Classification loss is introduced in Section 3.3.3.

For multi-label datasets, an affinity matrix is used as binary supervisory information to constrain feature similarity. Herein, all similar pairs are constrained equally [173, 178]. Each objective value in the affinity matrix is set to 1 if an image and text have at least one common label. However, similar image-text pairs may have different levels of similarity depending on the number of common labels they have.

In this chapter, we address above limitations by proposing a novel network, as shown in Figure 3.1. The novelty of this chapter is summarized as two-fold. First, we incorporate Shannon's information entropy [179] to directly map features for image and text into a common space where their heterogeneous modality properties are not exhibited. Specifically, given a hash code which corresponds to image or text, the network, after being trained well, will yield a high uncertainty with respect to modality the hash code belongs to. To the best of our knowledge, this work is the first to use information entropy [179] for cross-modal hash retrieval. Second, we propose a multi-level feature similarity which considers the number of common labels between similar image-text pairs to constrain these pairs with different weights.

3.2 Cross-modal Hash Learning

Recently, a variety of cross-modal hash learning methods are proposed to minimize the heterogeneity gap. Regarding supervised methods to improve retrieval performance, Jiang *et al.* [173] proposed DCMH to integrate deep feature learning and hash code learning into a unified structure where a similarity matrix was used as supervisory information. Aiming at learning a common latent space for image and text, Li *et al.* [178] introduced a three stream self-supervised hashing network where embedded features in a common space were used to predict semantic labels. For these methods, each similar image-text pair could be well projected as semantically-related feature vectors. However, the holistic feature distributions of two modalities

are still inconsistent (*i.e.* showing a heterogeneity gap). To mitigate this issue, adversarial learning methods are incorporated [175, 177, 178]. Chi *et al.* [175] introduced a dual structure for common representation learning in which new samples are generated via Generative Adversarial Networks (GANs) [180] and original ones are reconstructed. Their method can solve the problem of adding new categories in cross-modal retrieval; Wang *et al.* [177] introduced a feature projector and domain classifier which run as minimax game with adversarial learning, but the Gradient Reversal Layer (GRL) [181] and domain labels are needed in their approach.

We consider a holistic feature distribution in the common space and incorporate the information entropy [179] to maximize the uncertainty of visual and textual domains, such that modality properties are not exhibited, while preserving the semantic similarity of hash codes by using pair-wise and classification-based loss functions.

3.3 Domain Uncertainty Measurement via Information Theory

For the image-text dataset with n samples, we use $\mathbf{X} = \{x_i, l_i\}_{i=1}^n$ to denote the images and their labels, we use $\mathbf{Y} = \{y_i, l_i\}_{i=1}^n$ to denote the text and their labels. Here $l_i = [l_{i1}, l_{i2}, ..., l_{ic}]$ are multi-label annotations of images and text, and c is the total number of classes. We define a binary similarity matrix $\mathbf{S}$ where $S_{ij} = 1$ when x_i and y_i have at least one common label, otherwise $S_{ij} = 0$. Additionally, we define a multi-level similarity weight $w_{ij} = t_{ij}/c$ where t_{ij} is the number of common labels between x_i and y_i. Given these training data and a supervised matrix, the task of the cross-modal hash retrieval is to learn two *sign* functions for the two modalities: $B(x_i) = sign(F(x_i, \boldsymbol{\theta}_v)) \in \{-1, +1\}^K$, $B(y_i) = sign(G(y_i, \boldsymbol{\theta}_t)) \in \{-1, +1\}^K$, where K is the length of hash codes, $\boldsymbol{\theta}_v$ and $\boldsymbol{\theta}_t$ are the network parameters for feature learning for two modalities. According to the binary similarity matrix $\mathbf{S}$, similar pairs ($F(x_i)$, $G(y_i)$) should be represented by similar hash codes ($B(x_i)$, $B(y_i)$) in the Hamming space. Usually, as $B(\cdot)$ is a discrete function and it is not differentiate, a soft continuous relaxation $H(\cdot) = tanh(\cdot)$ is used to replace $B(\cdot)$.The hash code can be optimized using:

$$L_q = \left(\|\mathbf{H}^v - \mathbf{B}^v\|_F^2\right) + \left(\|\mathbf{H}^t - \mathbf{B}^t\|_F^2\right) \tag{3.1}$$

The aim of our method is to learn a better common space for real-valued features $\mathbf{F}(\cdot)$, $\mathbf{G}(\cdot)$ and hash codes $\mathbf{H}(\cdot)$, $\mathbf{B}(\cdot)$ where multi-level similarity degrees are also preserved. The whole framework is depicted in Figure 3.1.

3.3.1 Information theory and domain uncertainty

As shown in Figure 3.2(a), real-valued features extracted from visual and textual domains (F^I and G^T in Figure 3.1, respectively) are semantically similar but in-

consistently distributed. Samples from two domains have different domain-related properties. For example, textual data have more abstract semantics than visual data. These properties will often result in feature distributions which still hold this information giving higher certainty on the domain to which the input data belongs (*i.e.* the visual domain or textual domain). More specifically, when it is possible to identify a feature in the common space coming from the visual domain with higher probability (P_i) rather than coming from textual domain with lower probability ($P_t = 1 - P_i$), domain uncertainty is not achieved. Thus, for a given feature, it can not be determined which domain it originally belongs to, it means that this feature is identified from two domains with equal probability ($P_i = P_t = 0.5$), and the common space has highest uncertainty corresponding to highest information entropy. As in [179], we incorporate information entropy to measure the uncertainty of two domains. Figure 3.2(b) illustrates that two domains with equal probability leads to highest information entropy and information content.

Domain uncertainty is in proportional to information entropy [179], as shown in Figure 3.2(c). Based on this observation, we devise a domain uncertainty loss function using information entropy. When the objective function is minimized, the information entropy will be maximized, which means that the common space maximizes domain uncertainty. Specifically, we build domain predictor network $\mathcal{D}$ which includes three fully-connected (FC) layers. The output probability is $P_j^d(\cdot) = \mathcal{D}(\cdot, \boldsymbol{\theta}_d)$, "$\cdot$" indicates features from image or text shared with the parameter $\boldsymbol{\theta}_d$. The output neurons of prediction layer are $M.$:

$$\begin{aligned} \min \underbrace{(L_d^r + L_d^b)}_{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t, \boldsymbol{\theta}_d} = & \sum_{i=1}^{N} \sum_{j=1}^{M} \Big(P_{d,j}^r \big(\mathcal{F}(\cdot)\big) * log\big(P_{d,j}^r(\mathcal{F}(\cdot))\big) \\ & + P_{d,j}^b \big(\mathcal{H}(\cdot)\big) * log\big(P_{d,j}^b(\mathcal{H}(\cdot))\big) \Big) \\ s.t. \quad & \mathcal{F}(\cdot) = \mathbf{F}(x, \boldsymbol{\theta}_v) \ or \ \mathbf{G}(y, \boldsymbol{\theta}_t), \\ & \mathcal{H}(\cdot) = \mathbf{H}(x, \boldsymbol{\theta}_v) \ or \ \mathbf{H}(y, \boldsymbol{\theta}_t), \end{aligned} \tag{3.2}$$

where L_d^r is the loss component for the real-valued features used to predict domain probability and L_d^b indicates the loss component for the binary features used for domain prediction. N is the number of training samples, and M is set to 2, which denotes the number of domains in this task.

3.3.2 Multi-level feature preserving

A binary similarity matrix $\mathbf{S}$ can be used to preserve pair-wise similarity. Each $S_{ij} = 1$ when the corresponding image and text have at least one common label. However, similar image-text pairs may have different levels of similarity. Namely, different pairs can have different number of common labels, but the matrix $\mathbf{S}$ constrains these pairs equally. Considering this limitation of $\mathbf{S}$, we define a multi-level similarity matrix $\mathbf{W}$, which holds different similarity weights for all similar pairs. We depict

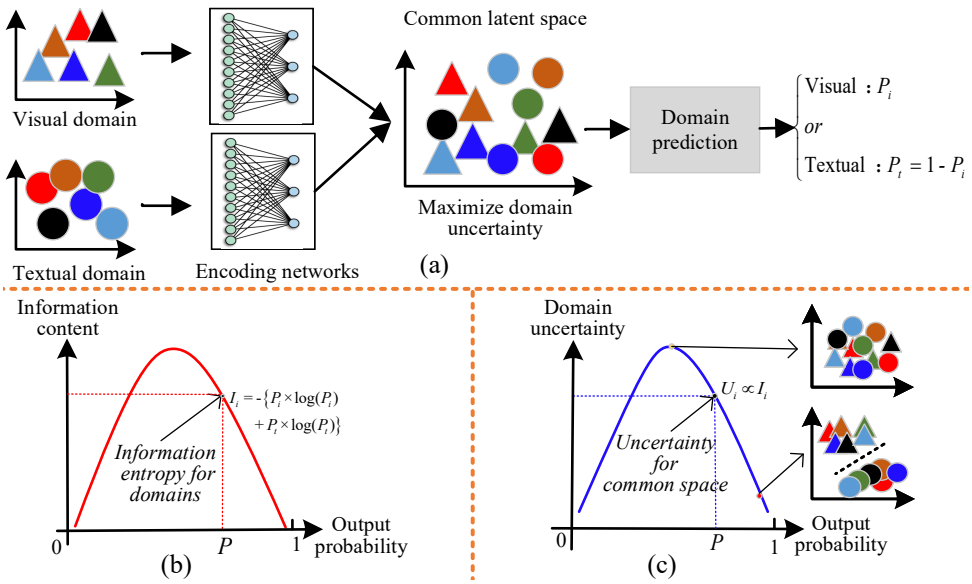


Figure 3.2: (a): Images and text are embedded via non-shared encoding sub-networks. The domain uncertainty can be predicted by using the output probabilities from a predictor. (b): Relationship between information entropy and predicted probability. (c): Relationship between domain uncertainty and output probabilities. When probabilities predicted for two modalities are identical, the shared space is intertwined into a domain confusion state (*i.e.* most uncertain). If one modality is identified with a higher probability (closer to 1) while another with a lower probability (closer to 0), the domain confusion state is not achieved.

the multi-level similarity matrix and binary similarity matrix in Figure 3.3. Each value w_{ij} in $\mathbf{W}$ is normalized by the total number of class in a dataset.

The real-valued features and binary features of image x_j are denoted as a triplet vector $\{F^{x_i}, H^{x_i}, B^{x_i}\}$, and the feature of a text y_j as triplet $\{G^{y_j}, H^{y_j}, B^{y_j}\}$. Then, $\mathbf{W}$ can be used to regularize a more specific similar pairs by using:

$$\min \underbrace{(L_m^r + L_m^b)}_{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t} = \sum_{i,j=1}^{N} \Big(\big(\delta(2\Delta_{ij}^r) - w_{ij}\big)^2 + \big(\delta(2\Gamma_{ij}^b) - w_{ij}\big)^2 \Big) \quad s.t. \quad w_{ij} = t_{ij}/c, \tag{3.3}$$

where L_m^r and L_m^b correspond to real-valued and binary features, $\delta(\cdot)$ is the sigmoid function, w_{ij} is the above defined multi-level similarity weight. $\Delta_{ij}^r = \frac{1}{2}(F_{*i})^T(G_{*j})$ and $\Gamma_{ij}^b = \frac{1}{2}(H_{*i})^T(H_{*j})$ denote the inner product of image and text features; H_{*i} and H_{*j} correspond to soften visual and textual hash codes, respectively.

As suggested in [173, 178], we also use the binary similarity matrix $\mathbf{S}$ to define the pair-wise objective function. Specifically, for S_{ij}, the conditional probability for each pair (F^{x_i}, G^{y_j}) and (H^{x_i}, H^{y_j}) can be computed by using:

$$p(S_{ij}|B) = \begin{cases} \delta(\psi_{ij}) & S_{ij} = 1, \\ 1 - \delta(\psi_{ij}) & S_{ij} = 0, \end{cases} \tag{3.4}$$

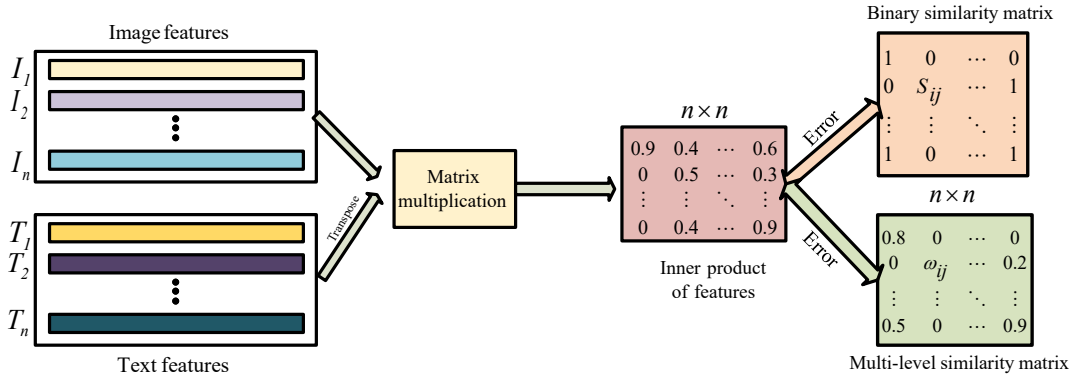


Figure 3.3: Multi-level similarity matrix and binary similarity matrix are used as supervised information for feature learning.

where $\delta(\psi_{ij})$ is the sigmoid function and ψ_{ij} is the inner product of input features. Pair-wise objective function is:

$$\min \underbrace{(L^r_{pairs} + L^b_{pairs})}_{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t} = \sum_{i,j=1}^{N} \Big(\big(S_{ij}\Delta^r_{ij} - log(1 + e^{\Delta^r_{ij}})\big) + \big(S_{ij}\Gamma^b_{ij} - log(1 + e^{\Gamma^b_{ij}})\big) \Big) \tag{3.5}$$

where Δ^r_{ij} and Γ^b_{ij} are set as in Eq. 3.3.

3.3.3 Classification-based objective function

Furthermore, as shown Figure 3.1, we build a label predictor $\mathcal{L}$ to output the probability $P^l_i(\cdot) = \mathcal{L}(\cdot, \boldsymbol{\theta}_l)$. We only use the length-fixed real-valued feature $F(\cdot)$ and $G(\cdot)$ for label prediction because the length of hash codes is changed. The objective function for the label prediction is defined as:

$$\min \underbrace{(L^v_l + L^t_l)}_{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t, \boldsymbol{\theta}_l} = -\sum_{i=1}^{N} \Big(l_i \cdot log(p^{v,t}_i) + (1 - l_i) \cdot log(1 - p^{v,t}_i) \Big) \tag{3.6}$$

where $p^v_i = \mathcal{L}(F(x_i), \boldsymbol{\theta}_l)$, $p^t_i = \mathcal{L}(G(y_i), \boldsymbol{\theta}_l)$ denote the sigmoid output probabilities of label predictor, l_i are the ground-truth labels. The dimension of p^v_i and p^t_i are equal to the number of labels in each dataset.

Finally, the global objective will be:

$$L = \alpha\ L_d + \beta\ L_{pairs} + \gamma\ L_l + \eta\ L_m + \epsilon\ L_q \tag{3.7}$$

3.4 Experiments and Evaluations

3.4.1 Implementation details

We utilize the CNN-F from [75] as the backbone network for visual feature learning. As shown in Figure 3.1, activations from *FC7* layer are projected into a common space using a 3 FC layer ($4096 \rightarrow K \rightarrow N$), where K is the dimension of the common feature. We use BoW to embed textual features and then adopt a multi-scale (MS) fusion FC layer ($\text{T} \rightarrow \text{MS} \rightarrow 2000 \rightarrow K \rightarrow N$) to learn the textual features. Following [178], MS fusion model has five-level pooling layers. The label predictor and domain predictor consist of 3 FC layers in which the number of neurons go from ($512 \rightarrow 256 \rightarrow c$) and ($512 \rightarrow 256 \rightarrow 2$), respectively, where c is 24 for the MIRFlickr-25K and 21 for the NUS-WIDE dataset. For all FC layers, we set the dropout rate to 0.9. For optimizing the network, we adopt the alternating learning strategy from [173] where we fine-tune visual parameters and fix textual parameters. Regarding to the hyperparameters in Eq.7, we analyze the parameter sensitivity, as reported in Figure 4. Based on these observations, we set $\alpha = 100, \beta = \gamma = \epsilon = 1, \eta = 0.1$. The learning rate varies from 10^{-4} to 10^{-8}.

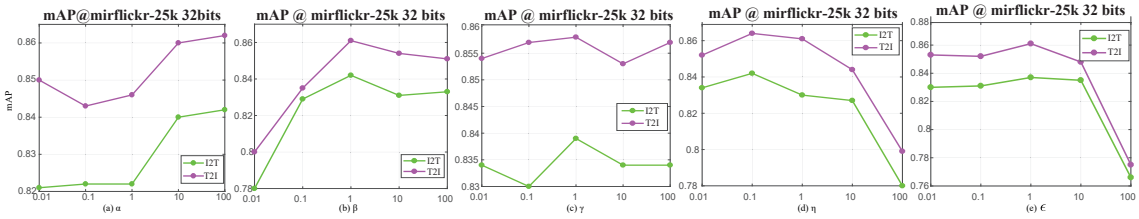


Figure 3.4: Sensitivity analysis of the hyperparameters in loss function in Eq. 3.7.

3.4.2 Datasets

The **MIRFLICKR-25K** [182] dataset contains 25,000 instances. We follow the experiment protocols given in [173]. In total, 20,015 image-text pairs are selected for our experiment. The text for each sample is embedded into a 1386 dimensional BoW representation. There are 24 labels for each pair. The number of training pairs is 10,000 and the number of query pairs is 2,000.

The **NUS-WIDE** [183] dataset contains 269,648 images. There are 81 ground-truth concepts that have been annotated manually. Following the protocols in [178], we select the 21 most frequent concepts as the training set (190,421 in total) in which the number of training samples is equal to 10,500 and query set has size of 2,100. Each annotation is embedded into a 1000 dimensional BoW representation.

3.4.3 Performance and evaluation

We adopt Hamming ranking and hash lookup to evaluate the performance. For hash based retrieval, the Hamming ranking procedure ranks the candidates in the retrieval

Table 3.1: mAP for different feature dimension N on the MIRFlickr-25k dataset.

Feature dimension $N =$	64	128	256	512	1024
Image-to-Text	0.802	0.818	0.831	**0.833**	0.829
Text-to-Image	0.819	0.850	0.852	**0.859**	0.844

set according to their Hamming distance to the given query items in ascending order. Mean average precision (mAP) is the commonly-used criteria to measure the accuracy of the Hamming ranking distances. The accuracy of the hash look-up returns all the candidates within a certain Hamming radius. A precision-recall curve is widely used to implement hash look-up evaluation. For performance comparison, we compare with recent relevant work in DCMH [173] and SSAH [178], both of which use deep learning methods.

Common feature dimension. The dimension of the common feature is an important parameter for cross-modal hash retrieval. Before conducting our experiments, we evaluate the effect of the common feature dimension (*i.e.* the N in Figure 3.1). The results are reported in Table 3.1 where "Image-to-Text" and "Text-to-Image" mean that the query items are image and text, respectively. We can see that for $N = 512$, the mAP score is highest. Therefore, in our experiments, we use a 512 dimensional (*i.e.* $N = 512$) common feature.

Hamming ranking. To demonstrate the precision of our proposed method, we conduct and compare methods using CNN-F features on the MIRFlickr-25k and NUS-WIDE, as shown in Table 3.2. The baseline results are from SSAH [178] and we find that our method outperforms these baseline methods. Specifically, the proposed method achieves better significantly results than other counterparts. For instance, when the length of the hash codes is equal to 32 bits, the results for "Image-to-Text" and "Text-to-Image" are improved by 5.4% and 8.1%, respectively, when compared to state-of-the-art method SSAH. Meanwhile, for another dataset NUS-WIDE, where more instances and contents are included within an image, which makes it hard to train and perform cross-modal retrieval. However, the proposed method also outperforms the other methods. For instance, our method has 2.5%

Table 3.2: mAP results on MIRFlickr-25k and NUS-WIDE datasets.

Tasks and Methods		MIRFlickr-25K			NUS-WIDE		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Image-to-Text	DCMH [173]	0.735	0.737	0.750	0.478	0.486	0.488
	SSAH [178]	0.782	0.790	0.800	0.642	0.636	0.639
	Ours	**0.825**	**0.833**	**0.838**	**0.648**	**0.652**	**0.647**
Text-to-Image	DCMH [173]	0.763	0.764	0.775	0.638	0.651	0.657
	SSAH [178]	0.791	0.795	0.803	0.669	0.662	0.666
	Ours	**0.845**	**0.859**	**0.861**	**0.671**	**0.681**	**0.669**

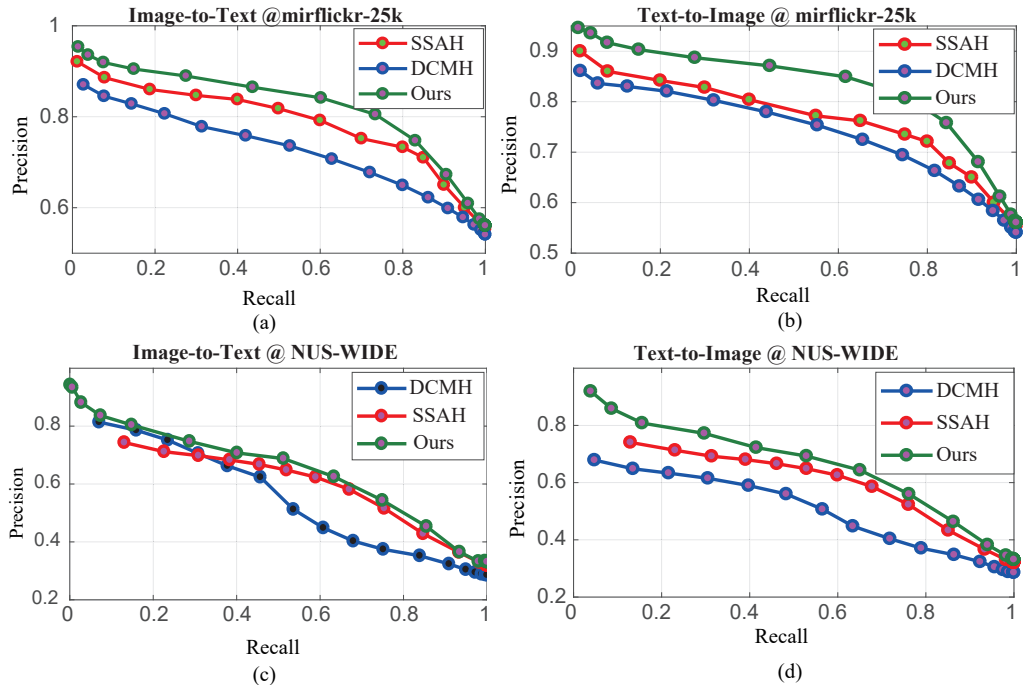


Figure 3.5: Precision-recall curves for three methods. The code length is 16 bits.

Table 3.3: Ablation study for the proposed method.

Tasks	Image-to-Text	Text-to-Image
Baseline1	0.773	0.792
Baseline2	0.795	0.814
Baseline3	0.810	0.827
Full-method	0.834	0.859

and 2.9% improvement respectively, compared to SSAH using a hash codes of 32 bits. Therefore, all the results in Table 3.2 demonstrate the effectiveness of using information entropy for mitigating the heterogeneity gap. Furthermore, we could find that for different tasks and using a different hash code length, we can find the retrieval performance improves when the hash code length is set to 32 bits.

Hash lookup. For this procedure, we compute the precision and recall for the retrieval results with respect to a different Hamming radius. In this experiment, we vary the Hamming radius from 0 to 50 with step-size 1. For each radius, the retrieval algorithms will return the correct items, larger covered area of the precision-recall curve indicates a better retrieval performance. The results are shown in Figure 3.5. For fair comparison, we used the source codes provided by the authors, and a hash code length of 16 bits for this experiment. For both the "Image-to-Text" and "Text-to-Image" tasks, our proposed method has curves that have a larger covered areas than these competitive deep learning methods. The result further demonstrates the superiority of the proposed method.

Ablation study. We conduct an ablation study for our method on the MIRFlickr-

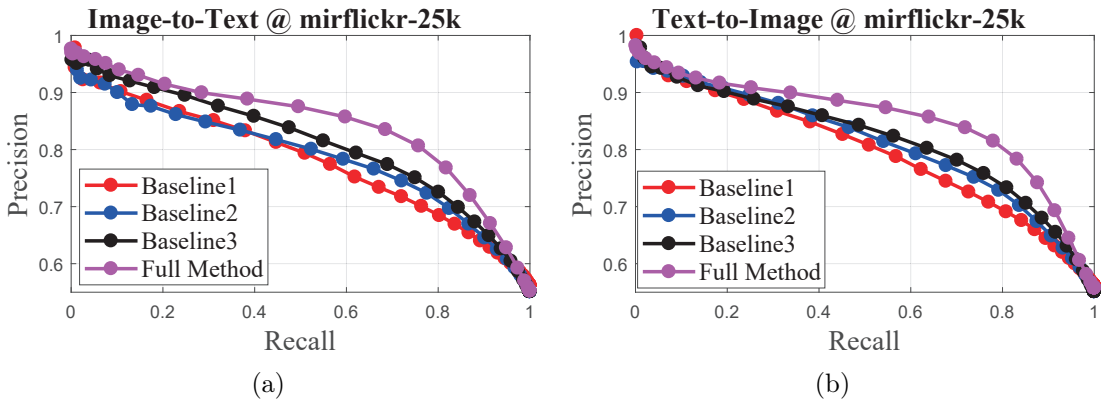


Figure 3.6: Precision-recall curves for ablation study. The code length is 32 bits.

25k dataset with 32-bits hash codes. Specifically, we build three baselines using different objective functions. Our *Baseline1* is only based on $L_{pairs} + L_q$; *Baseline2* is based on $L_{pairs} + L_q + L_l$, which illustrates the effectiveness of label predictor; *Baseline3* is based on $L_{pairs} + L_q + L_l + L_m$, demonstrating the effect of a multi-level similarity objective function. Finally, we incorporate all loss functions as *full-method*. The results are reported in Table 3.3. Furthermore, we compare the corresponding precision-recall curves, as shown in Figure 3.6. We can see that the mAP is highest when domain uncertainty is used.

3.5 Chapter Conclusions

In this chapter, we have exploited modality information for cross-modal hash retrieval. We devised a novel network to predict visual domain and textual domain based on the features learned from these two modalities. The protagonist network depends on a objective function by using Shannon's information entropy to maximize domain uncertainty. Maximizing the domain uncertainty is beneficial for bridging the gap between two modalities because it minimizes the influence of the individual modality. Furthermore, we considered multi-level similarity for feature learning where all similar image-text pairs are constrained with different weights according to the number of common labels between these similar pairs. Extensive experiments implemented on two multi-label datasets demonstrate the effectiveness of the proposed method which outperforms the state-of-the-art.

Chapter 4

Integrating Information Theory and Adversarial Learning for Cross-modal Retrieval

In this chapter, we further explore cross-modal retrieval to address the challenges posited by the heterogeneity gap and the semantic gap. To be specific, we propose integrating Shannon information theory and adversarial learning. In terms of the heterogeneity gap, we integrate modality classification and information entropy maximization adversarially. For this purpose, a modality classifier (as a discriminator) is built to distinguish the text and image modalities according to their different statistical properties. This discriminator uses its output probabilities to compute Shannon information entropy, which measures the uncertainty of the modality classification it performs. Moreover, feature encoders (as a generator) project uni-modal features into a commonly shared space and attempt to fool the discriminator by maximizing its output information entropy. Thus, maximizing information entropy gradually reduces the distribution discrepancy of cross-modal features, thereby achieving a domain confusion state where the discriminator cannot classify two modalities confidently. To reduce the semantic gap, Kullback-Leibler (KL) divergence and bi-directional triplet loss are used to associate the intra- and inter-modality similarity between features in the shared space. Furthermore, a regularization term based on KL-divergence with temperature scaling is used to calibrate the biased label classifier caused by the data imbalance issue.

Keywords

Cross-modal retrieval, Shannon information theory, Adversarial learning, Modality uncertainty, Data imbalance.

This chapter is based on the following publication [35]:

- Chen, W., Liu, Y., Bakker, E., and Lew, M.S., "Integrating Information Theory and Adversarial Learning for Cross-modal Retrieval." Pattern Recognition, 2021, 117, pp. 107983.

4.1 Introduction

Deep learning methods can effectively embed features from different modalities into a commonly shared space, and then measure the similarity between these embedded features. As mentioned in Chapter 3, the "heterogeneity gap" [176] and the "semantic gap" [10] are still challenges to be addressed for cross-modal retrieval. To achieve better retrieval performance, it is essential to address these gaps for associating the similarity between cross-modal features in the shared space.

To capture the semantic similarity between cross-modal features, many approaches have been proposed in recent years. Some approaches focus on designing effective structures from a deep networks perspective. For instance, graph convolutional networks are employed to model the dependencies within visual or textual data. Other approaches focus on designing similarity constraint functions from a deep features perspective. For example, bilinear pooling-based methods are applied to align image and text features to then accurately capture inter-modality semantic similarity. In other examples, coordinated representation learning methods, such as ranking loss [177, 184] are widely used to preserve similarity between cross-modal features. These constraint functions mainly aim at reducing the semantic gap by focusing on the similarity between two-tuple or three-tuple samples. However, they might not directly mitigate the heterogeneity gap caused by the inconsistent feature distributions in the different spaces.

Considering the limitations of similarity constraint functions, we propose a new method to perform cross-modal retrieval from two aspects. First, we reduce the heterogeneity gap by integrating Shannon information theory [179] with adversarial learning, in order to construct a better embedding space for cross-modal representation learning. Second, we combine two loss functions, including KL-divergence loss and bi-directional triplet loss, to preserve semantic similarity during the feature embedding procedure, thereby reducing the semantic gap.

To do this, we combine the information entropy predictor and the modality classifier in an adversarial manner. Information entropy maximization and modality classification are two processes trained with competitive goals. Since uni-modal features extracted from image or text data are characterized by different statistical properties, it can be used to distinguish the original modalities these features belong to. As a result, when these features in the shared space are correctly classified into their original modalities with high confidence, then their feature distributions convey less information content, and the modality classifier performs modality classification with lower uncertainty. In contrast, when cross-modal features become modality-invariant and show their commonalities, these features cannot be classified into the modality they originally belong to. In this case, the feature distributions in the shared space conveys more information content and higher modality uncertainty.

According to Shannon's information theory [179], we can measure the modality uncertainty in the shared space by computing information entropy. This basic proportional relation provides the principle to mitigate the heterogeneity gap. For this purpose, we integrate modality uncertainty measurement into cross-modal representation learning. As shown in Figure 4.1, a modality classifier (in the following we call it a *discriminator*) is devised to classify image and text modality, rather than perform a "true/false" binary classification. This discriminator also provides its output probabilities to calculate the information entropy of the cross-modal feature distributions. At the start of training, the discriminator can classify images and text modalities with high confidence due to their different statistical properties. In contrast, the feature encoders (in the following we call it a *generator*) project features into a shared space and attempt to fool the discriminator and make it perform an incorrect modality classification until features in the shared space are fused heavily into a confusion state, maximizing the modality uncertainty.

On the basis of this heavily-fused state, we further use similarity constraints on the feature projector to reduce the semantic gap. Specifically, KL-divergence loss is used to preserve semantic similarity between image and text features by using instance labels as supervisory information. More importantly, we consider the issue of data imbalance and introduce a regularization based on KL-divergence with temperature scaling to calibrate the biased label classifier. Afterwards, we adopt the commonly used bi-directional triplet loss and instance label classification loss (*i.e.* categorical cross-entropy loss) to achieve good retrieval performance.

4.2 Related Work

4.2.1 Cross-modal representation learning and matching

Preserving the similarity between cross-modal features should consider two aspects: inter-modality and intra-modality. Supervision information (*e.g.* class label or instance label), if available, is beneficial for learning features from these two aspects. Preserving feature similarity can be realized by using methods such as joint representation learning and coordinated representation learning. Joint representation learning methods project the uni-modal features into the shared space using straightforward strategies such as feature concatenation, summation, and inner product. Subsequently, more complicated bilinear pooling methods, such as multimodal compact bilinear (MCB) pooling, are proposed to explore the semantic similarity of cross-modal features. To regularize the joint representations, deep networks are commonly trained by using objective functions, such as regression-based loss [185].

Coordinated representation learning methods process image and text features separately but impose them under certain similarity constraints. In general, these constraints can be categorized into classification-based and verification-based methods in supervised scenarios. In terms of classification-based methods, both image

and text features are used to make a label classification by using categorical cross-entropy loss function. Because a paired image-text input has the same class label, their features can be associated in the shared space. However, classification-based methods cannot preserve the similarity between inter-modality features well because the similarity between image and text features is not directly regularized.

Verification-based methods, based on metric learning, are proposed to further optimize inter-modality feature learning. Given a similar (or dissimilar) image-text pair, their corresponding features should be verified as similar (or dissimilar). Therefore, the goal of deep networks is to push features of similar pairs closer, while keeping features of dissimilar pairs further apart. Verification-based methods include pair-wise constraints and triplet constraints, which focus on inferring the matching scores of image-text feature pairs [185].

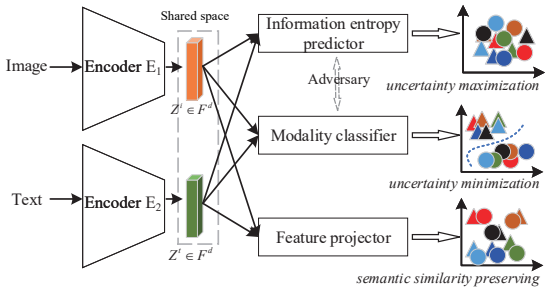


Figure 4.1: Illustration of combining information theory and adversarial learning. The features $Z^i \in F^d$ and $Z^t \in F^d$ with dimension d for image-text pairs are extracted using deep neural networks. Shape indicates modality and color denotes pair-wise similarity information.

Triplet constraints optimize the distance between positive pairs to be smaller than the distance between negative pairs by a margin. They can capture both intra-modality and inter-modality semantic similarity. For example, bi-directional triplet loss has been employed to optimize image-to-text and text-to-image ranking [177]. Although triplet constraints are widely used for cross-modal retrieval, the difficulties are in the mining strategy for negative pairs and the selection of a margin value, which are usually task-specific and empirically selective.

4.2.2 Adversarial learning for cross-modal retrieval

The aforementioned joint and coordinated representation learning approaches focus on two-tuple or three-tuple samples, which may be insufficient for achieving overall good retrieval performance. Adversarial learning, as an alternative method, has shown its powerful capability for modeling feature distributions and learning discriminative representations between modalities when deep networks are trained with competitive objective functions [177].

Recent progress in using adversarial learning for cross-modal retrieval can be categorized as feature-level and loss function-level discriminative models.

From a feature-level perspective, it is possible to preserve semantic consistency by performing a min-max game between inter-modality feature pairs [177]. A straightforward way is to build a discriminator, making a “true/false” classification between

image features (regarded as true), corresponding matched text features (regarded as fake), and unmatched image features from other categories (also regarded as fake) [177]. Alternatively, a cross-modal auto-encoder can be combined to generate features for another modality. For example, a generator attempts to generate image features from textual data and then regards them as true, while for a discriminator, image features extracted from original images and these from the generated "images" are labeled as true and fake, respectively. The adversarial training explores the semantic similarity of cross-modal representations. Intra-modality discrimination also can be considered in cross-modal adversarial learning, forcing the generator to learn more discriminative features. In this case, the discriminator tends to discriminate the generated features from its original input.

From a loss function-level perspective, instead of making a binary classification (*i.e.* true or fake), adversarial learning is used to train two groups of loss functions or two processes with competitive goals. This idea is applied in recent work for cross-modal retrieval [177]. Specifically, a feature projector is trained to generate modality-invariant representations in the shared space, while a modality classifier is constructed to classify the generated representations into two modalities. Similarly, we combine two networks and train them with two competitive goals.

4.2.3 Information-theoretical feature learning

As noted before, feature vectors from different modalities are distributed in different spaces, resulting in the heterogeneity gap, which affects the accuracy of cross-modal retrieval. Therefore, it becomes essential to reduce feature distribution discrepancies and thereby reduce the heterogeneity gap. The solution for this is to measure and then minimize distribution discrepancy. For example, distribution disparity of cross-modal features can be characterized by Maximum Mean Discrepancy (MMD), which is a differentiable distance metric between distributions. However, MMD suffers from sensitive kernel bandwidth and weak gradients during training.

Information-theoretical based methods measure the differences of feature distributions and learn better cross-modal features. As an example, the cross-entropy loss function is widely used to estimate the errors between inference probabilities and ground-truth labels where the gradients are computed according to the errors. Once the gradients are computed, deep networks can further update their parameters via the back-propagation algorithm. KL-divergence (also called relative entropy) is another popular criterion to characterize the difference between two probability distributions. Minimizing the difference is beneficial for retaining the semantic similarity between features. For example, Zhang *et al.* [186] employ the KL-divergence to measure the similarity between projected features and supervisory information.

Recently, Shannon information entropy [179] has been used for performing cross-modal hash retrieval [34]. This study indicates that Shannon entropy can be used for

multimodal representation learning by estimating uncertainty [179]. Take generative adversarial networks as an example: if the generator makes image features and text features close and minimizes their discrepancy, then the discriminator will become less-certain or under-confident, *i.e.*, having a high information entropy to predict which modality each feature comes from. We applied this principle in our previous work [34] to design an objective function to maximize the domain uncertainty over cross-modal hash codes in a commonly shared space. Deep networks trained by using information entropy construct a domain confusion state where the heterogeneity gap can be effectively reduced. On the basis of this state, other loss functions, such as ranking loss, can be further applied to regularize feature similarity.

4.3 Proposed Approach

4.3.1 Problem formulation

We consider a supervised scenario for cross-modal retrieval. Denote X^i as the input images and the corresponding descriptive sentences as X^t. Each image and its descriptive sentences have the same instance label Y. Therefore, we can organize an input pair (x^i, x^t, y) to train a deep network. To be specific, feature encoders $E_1(\cdot;\boldsymbol{\theta}_{E_1})$ and $E_2(\cdot;\boldsymbol{\theta}_{E_2})$ extract image and text features, respectively, and then further embed these uni-modal features into a shared space by using non-shared sub-networks. The embedded features with dimension d are denoted as $Z^i = E_1(X^i;\boldsymbol{\theta}_{E_1})$ and $Z^t = E_2(X^t;\boldsymbol{\theta}_{E_2})$, Z^i,$Z^t \in R^d$. Note that the parameters in the non-shared sub-networks for uni-modal image and text feature embedding have been included into $\boldsymbol{\theta}_{E_1}$ and $\boldsymbol{\theta}_{E_2}$, respectively. The goal is to train a deep network to make the embedded features Z^i and Z^t modality-invariant and semantically discriminative, improving the retrieval accuracy.

As shown in Figure 4.1, the networks E_1, E_2, and the information entropy predictor act as a generator, while the modality classifier acts as a discriminator. The training of the generator and the discriminator is formulated as an min-max game to mitigate the heterogeneity gap. The feature projector preserves feature similarity under several constraints, which are introduced in Section 4.4.2, 4.4.3, and 4.4.4.

4.3.2 Integrating information theory & adversarial learning

4.3.2.1 Information entropy and modality uncertainty

Uni-modal features from different modalities have similar semantics but are distributed in different spaces. Their similarities are not well associated so that these features are not directly comparable. It is required to further embed them into a shared space (*i.e.* Z^i and Z^t in Figure 4.1). Uni-modal features are characterized by different statistical properties. Therefore, as shown in Figure 3.2(a) in Chapter 3, it is possible to identify a feature in the shared space coming from a visual modality

with higher probability P_i (more certain classification) than coming from a textual modality with lower probability $P_t = 1 - P_i$ (less certain classification). In other words, these cross-modal features are not intertwined heavily. As a result, the domain confusion state is not achieved. Conversely, if a given feature can not be distinguished which modality this feature originally comes from, it indicates that this feature has identical probability ($P_i = P_t$) coming from each modality. In this case, the shared space has highest uncertainty and the cross-modal features are intertwined into a domain confusion state, which corresponds to highest information content. We use information entropy [179] to measure the uncertainty of the shared space. Figure 3.2(b) in Chapter 3 illustrates that two modalities with an equal probability leads to the highest Shannon information entropy and thus information content.

Modality uncertainty refers to the unreliability of classification that the discriminator classifies image features and text features into two modalities. It is proportional to Shannon information entropy [179], as shown in Figure 3.2(c) in Chapter 3. Based on this observation [34], we design the discriminator to measure its output modality uncertainty by using information entropy as a criterion. Maximizing information entropy means that the discriminator becomes least-confident in classifying the original modality of image and text features, resulting in the greatest reduction of the heterogeneity gap.

4.3.2.2 Adversarial learning and information entropy

To make cross-modal features modality-invariant, we devise a generator and a discriminator, as shown in Figure 4.1. The discriminator performs modality classification to identify visual modality and textual modality based on cross-modal features. Following [177], we define the modality label as Y_c^* for these two modalities (for visual modality $* = i$ and textual modality $* = t$). Using output probabilities of the discriminator, we can compute cross-entropy loss to realize modality classification [177]. Once the network convergences under the constraint of this loss function, visual modality and textual modality are clearly identified and classified, thereby minimizing the modality uncertainty.

Conversely, the generator is designed to maximize the modality uncertainty over the cross-modal feature distributions. To achieve this, the generator learns modality-invariant features to fool the discriminator, maximizing the uncertainty of modality classification the discriminator performs. If the modality uncertainty is maximized, the discriminator is most likely to make an incorrect modality classification and be least-confident about its classification results. In this case, cross-modal features are intertwined into a domain confusion state and become indistinguishable.

To this end, we explore the ways to integrate information entropy and adversarial learning into an end-to-end network, which is introduced in Section 4.4.1. For better understanding, we also explore another combining paradigm in the Experimental Section.

4.3.3 KL-divergence for cross-modal feature projection

To reduce the semantic gap, we use KL-divergence to characterize the differences between projected cross-modal features (Z^i and Z^t in Figure 4.1) and a supervisory matrix computed from their instance labels, *i.e.* $KL((f(Z^i, Z^t)||f(Y_l^\top, Y_l))$, (see Eq. 4.9). In this way, the semantic similarity among cross-modal features can be preserved. We illustrate this process in Figure 4.2. It is important to note that when using KL-divergence to preserve semantic similarity of cross-modal features, all positive and negative pairs in a mini-batch are considered. As for the supervisory matrix $f(Y_l^\top, Y_l)$, it is computed by using matrix multiplication and is normalized to the range from 0 and 1.

We argue that different operations to realize $f(Z^i, Z^t)$ affect similarity preserving. Directly, the operation $f(\cdot)$ can be an inner product on cross-modal features Z^i and Z^t. However, using the inner product has some implicit drawbacks. First, when multiplying one image feature vector with all text feature vectors, the results of the inner product are not optimally comparable due to the non-normalized text features, and vice versa. Second, the angles between each image feature vector and each text feature vector, as well as their whole feature distributions, are changing when training the deep network, which makes it problematic for an inner product to measure feature similarity.

To tackle the above limitations, we adopt a cross-modal feature projection to characterize the similarity between features. The idea is related to the work in [186]. Cross-modal feature projection is based on the same distribution and operates on the normalized features. For instance, an image feature vector, $z_j^i \in Z^i$, can be projected to the distribution of a text feature vector $z_k^t \in Z^t$, then each projected feature vector from image to text (termed "$i \rightarrow t$") can be formulated as:

$$\begin{aligned} \hat{z}_j^{i\rightarrow t} &= |z_j^i| * \frac{<z_j^i, z_k^t>}{|z_j^i||z_k^t|} * \frac{z_k^t}{|z_k^t|} \\ &= <z_j^i, \bar{z}_k^t> * \bar{z}_k^t \end{aligned} \tag{4.1}$$

where "i" and "t" represent the visual and the textual modality, respectively, "j" and "k" represent the index of each image feature and text feature in the shared space, respectively, $\bar{z}_k^t$ denotes the normalized feature. Therefore, the length of $\hat{z}_j^{i\rightarrow t}$ is equal to $|\hat{z}_j^{i\rightarrow t}| = |<z_j^i, \bar{z}_k^t>|$, and denotes the similarity between image feature z_j^i and text feature z_k^t. When associating each image feature z_j^i with all text features Z^t, we obtain all different lengths, Therefore, when projecting all image features into all text features Z^t, we get a similarity matrix $A_{i\rightarrow t}$, which is formulated as

$$A_{i\rightarrow t}(Z^i, Z^t) = \sum_{j=1}^{N}\sum_{k=1}^{N} |<z_j^i, \bar{z}_k^t>| = Z^i(\bar{Z}^t)^\top \tag{4.2}$$

Similarly, if projecting all text features into all image features Z^i, we obtain another similarity matrix $A_{t\to i}$:

$$A_{t\to i}(Z^t, Z^i) = \sum_{k=1}^{N}\sum_{j=1}^{N} |<z_k^t, \bar{z}_j^i>| = Z^t(\bar{Z}^i)^\top \tag{4.3}$$

In the above two equations, Z^i and Z^t represent the cross-modal features from two modalities. N is the number of samples in a mini-batch. These two similarity matrices are normalized by a softmax function. Afterwards, we use KL-divergence to characterize the difference between the normalized matrices and the supervisory matrix, *i.e.* $KL((f(Z^i, Z^t)||f(Y_l^\top, Y_l))$. The specific objective function is introduced in Section 4.4.2.

4.4 Implementation and optimization

We introduce the implementation and optimization of our proposed approach in this section. We employ four convolutional neural networks such as ResNet-152 [13] and MobileNet [187] to obtain image features and a Bi-directional LSTM (Bi-LSTM) [188] to extract text features. All the extracted image and text features are uni-modal. Later, we borrow the protocols of non-shared encoding sub-networks (fully-connected layers) in [186] to get the cross-modal features Z^i and Z^t.

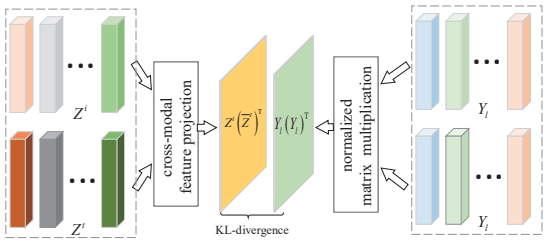


Figure 4.2: KL-divergence for cross-modal feature projection, which considers all features Z^i and Z^t in the shared space. Each paired image feature and text feature share the same instance label, indicated by the same color. The cross-modal feature projection module is critical to explore the similarity between image features and normalized text features. The projection process is formulated in Eqs. 4.2 and 4.3.

Once the cross-modal features are obtained, we use the proposed algorithm to train the networks based on the above theoretical analysis. The algorithm includes combining information entropy and adversarial learning to mitigate the heterogeneity gap, and loss function terms (*i.e.* KL-divergence loss, categorical cross-entropy loss, and bi-directional triplet loss) to preserve semantic similarity between cross-modal features.

4.4.1 Combining information theory & adversarial learning

We combine information entropy predictor and modality classifier in Figure 4.1 into a unified sub-network, as shown in Figure 4.3. In this paradigm, the discriminator D with parameters $\boldsymbol{\theta}_D$ performs a modality classification and computes the Shannon

information entropy. The backbone nets E_1 and E_2 for feature extraction act as the generator G. The whole structure forms a generative adversarial network. The information entropy computed from the discriminator back-propagates to the feature encoders. Specifically, when the discriminator is fixed, and its parameters are $\boldsymbol{\theta}_D^\star$, then the information entropy $H(P_D^\star) = \mathbb{E}_{i,t}(-P_D^\star * log(P_D^\star))$ is computed from its output probabilities $P_D^\star(D|Z^{i,t};\boldsymbol{\theta}_D^\star)$ across the features for all classes. Based on the information entropy, we can design a negative entropy loss $L_s = -H(P_D^\star)$ (see Eq. 4.4) to train the network. The gradients computed from L_s update the parameters of feature extractors. The negative information entropy L_s is label-free during training, and it regularizes the whole feature distribution to be modality-invariant.

The discriminator consists of some fully-connected layers. The last layer with two neurons yields probabilities that correspond to two modalities. This discriminator classifies whether the input features Z^i and Z^t are from the visual or the textual modality given the pre-defined modality label Y_c^*. In contrast, the generator (*i.e.* E_1 and E_2) aims at learning modality-invariant features to fool the discriminator to make an incorrect modality classification so that the generator gradually maximizes the output information entropy from the discriminator. Therefore, the learning process of the discriminator affects that of the generator in an indirect way. The objective function is calculated using the output probabilities $P_D(D|Z^{i,t};\boldsymbol{\theta}_D)$ of the discriminator.

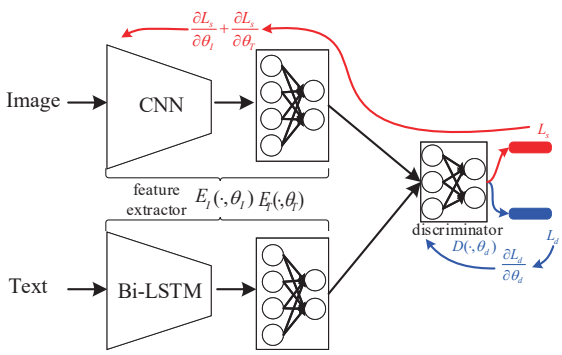


Figure 4.3: The implementation of integrating information entropy predictor and modality classifier in Figure 4.1 into a unified discriminator. Together with the feature extractors, the whole framework is in the form of generative adversarial network. For clarity, we ignore the feature projector, which includes label classification loss, bi-directional triplet loss, and KL-divergence loss.

For the generator E_1 and E_2:

$$
\begin{aligned}
L_s = & \frac{1}{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\Big(P_{D,m}^{i}(D^i|Z_j^i;\boldsymbol{\theta}_D)*log(P_{D,m}^{i}(D^i|Z_j^i;\boldsymbol{\theta}_D)) \\
& + P_{D,m}^{t}(D^t|Z_j^t;\boldsymbol{\theta}_D)*log(P_{U,m}^{t}(D^t|Z_j^t;\boldsymbol{\theta}_D))\Big) \qquad (4.4) \\
& s.t. \sum_{m=1}^{M} P_{D,m}^{*}(D^*|Z_j^*;\boldsymbol{\theta}_D) = 1,\ P_{D,m}^{*}(D^*|Z_j^*;\boldsymbol{\theta}_D) \geq 0
\end{aligned}
$$

It is expected for the generator G to maximize the information entropy $H(P_D^\star)$, and subsequently the modality uncertainty (see Figure 3.2 in Chapter 3). Since L_s is a negative entropy ($L_s = -H(P_D^\star)$) to maximize $H(P_D^\star)$, it is minimized to optimize

the parameters $\boldsymbol{\theta}_{E_1}$ and $\boldsymbol{\theta}_{E_2}$ of the generator during training. For the discriminator D, depending on the modality label Y_c^i and Y_c^t and its output probabilities $P_D(D|Z^{i,t};\boldsymbol{\theta}_D)$, the modality classification cross-entropy loss function is formulated as:

$$L_c = -\frac{1}{N}\sum_{j=1}^{N}\Big(Y_c^i * log\big(P_D^i(D^i|Z_j^i;\boldsymbol{\theta}_D)\big) + Y_c^t * log\big(P_D^t(D^t|Z_j^t;\boldsymbol{\theta}_D)\big)\Big) \tag{4.5}$$

L_c refers to the negative cross-entropy loss of the discriminator and is minimized to clearly classify image and text features into two modalities during training. Note that the gradients calculated from term L_s are only used to optimize the parameters $\boldsymbol{\theta}_{E_1}$ and $\boldsymbol{\theta}_{E_2}$ of the generator, whereas the gradients from term L_c are only for optimizing the parameters $\boldsymbol{\theta}_D$ of the discriminator, as shown in Figure 4.3. Minimizing loss L_c and L_s when trained iteratively will reduce the heterogeneity gap. The optimization method is straightforward, even though the gradients calculated from L_c will not directly affect the parameters of the feature encoders E_1 and E_2. The output probabilities of the discriminator change when updating its parameters, which will affect the Shannon information entropy and affect the output features from E_1 and E_2 in the end.

4.4.2 KL-divergence for similarity preserving

We also compute KL-divergence directly across Z^i and Z^t to further preserve semantic similarity. KL-divergence focuses on the projections of image and text features and is computed by $L_{kl} = KL((f(Z^i, Z^t)||f(Y_l^\top, Y_l))$. Here, superscript "$\top$" means matrix transpose. L_{kl} focuses on constraining the whole feature distributions and is complementary to the following bi-directional triplet loss function. We have introduced the process of cross-modal feature projection in Section 4.3.3. Given the similarity matrices (*i.e.* $A_{i\to t}(Z^i, Z^t)$ and $A_{t\to i}(Z^t, Z^i)$), we use the softmax function to normalize these matrices in Eq. 4.6 and Eq. 4.7. The supervisory matrix is normalized after matrix multiplication as in Eq. 4.8. Similar to [186], since we project features from visual (or textual) modality into textual (or visual) modality, the KL-divergence regularizes the semantics in bi-directional feature projection, which is formulated in Eq. 4.9 as:

$$P_{i\to t} = \frac{exp\big(A_{i\to t}(Z^i, Z^t)\big)}{\sum exp\big(A_{i\to t}(Z^i, Z^t)\big)} \tag{4.6}$$

$$P_{t\to i} = \frac{exp\big(A_{t\to i}(Z^t, Z^i)\big)}{\sum exp\big(A_{t\to i}(Z^t, Z^i)\big)} \tag{4.7}$$

$$Q_y = \frac{exp(Y_l^\top Y_l)}{\sum exp(Y_l^\top Y_l)} \tag{4.8}$$

$$\begin{aligned} L_{kl} &= L_{kl_{i\to t}} + L_{kl_{t\to i}} \\ &= \frac{1}{N}\Big(\sum\sum P_{i\to t} * log(\frac{P_{i\to t}}{Q_y+\varepsilon}) + \sum\sum P_{t\to i} * log(\frac{P_{t\to i}}{Q_y+\varepsilon})\Big) \end{aligned} \tag{4.9}$$

where ε is a small constant to avoid division by zero. Loss L_{kl} refers to the KL-divergence between the projections of image-text features and their supervisory matrix. This loss is minimized and the gradients computed from L_{kl} are used to update the parameters $\boldsymbol{\theta}_{E_1}$ and $\boldsymbol{\theta}_{E_2}$ of the generator, thereby the semantics between image features and text features can be associated.

4.4.3 Instance label classification

4.4.3.1 Categorical cross-entropy loss

Label classification is a popular idea for cross-modal features learning [186]. We use the instance labels provided on the datasets for label classification. For categorical cross-entropy loss, we apply the norm-softmax strategy and feature projection in [186] to learn more discriminative cross-modal features. On the one hand, the normalized parameters $\boldsymbol{\theta}_P$ in the label classifier encourage cross-modal features to distribute more compactly so that the softmax classifier performs label classification correctly. On the other hand, projection between image and text features strengthens their similarity association and is beneficial for label classification [186]. Feature projection can be computed using Eq. 4.1. Subsequently, given the instance label y_l, categorical cross-entropy loss L_{ce} is defined by Eq. 4.10 and is minimized during training[1]:

$$\begin{aligned} L_{ce} &= \mathbb{E}_{i,t}(-y_l * log(p_P(c|Z^{i,t};\boldsymbol{\theta}_P))) \\ &= -\frac{1}{N}\Big(\sum_{j=1}^{N} y_{l,j} * log\Big(\frac{exp(\mathbf{W}_{y_{l,j}}^\top \hat{z}_j^{i\to t})}{\sum_j exp(\mathbf{W}_j^\top \hat{z}_j^{i\to t})}\Big) + \sum_{j=1}^{N} y_{l,j} * log\Big(\frac{exp(\mathbf{W}_{y_{l,j}}^\top \hat{z}_j^{t\to i})}{\sum_j exp(\mathbf{W}_j^\top \hat{z}_j^{t\to i})}\Big)\Big) \\ & s.t. \quad ||\mathbf{W}_j|| = 1; \hat{z}_j^{i\to t} = <z_j^i, \bar{z}_j^t> * \bar{z}_j^t; \hat{z}_j^{t\to i} = <z_j^t, \bar{z}_j^i> * \bar{z}_j^i \end{aligned} \tag{4.10}$$

where N is the number of image-text pairs in a mini-batch. $W_{y_{l,j}}$ and W_j represent the $y_{l,j}$-th and the j-th column of weights $\mathbf{W}$ in classifier parameters $\boldsymbol{\theta}_P$ according to [186]. $\hat{z}_j^{i\to t}$ and $\hat{z}_j^{t\to i}$ are the projections image to text and the projections text to image, respectively, by using Eq. 4.1.

4.4.3.2 KL-divergence for data imbalance

Label classification using categorical cross-entropy loss can preserve semantic similarity between cross-modal features. However, we argue that there also exists a data imbalance issue when training the label classifier because each image is described

[1]We omit the bias term for simplicity

by more than one sentence (*e.g.* each image has five description sentences in the Flickr30K dataset). In the end, it causes the learned label classifier to prefer text features.

The issue of data imbalance in cross-modal retrieval can be resolved by constructing an augmented semantic space to re-align features. In this work, we use the temperature scaling [189] to tackle the data imbalance issue. The biased label classifier can be calibrated by re-scaling its output probabilities *i.e.*, $p^{i\to t}{=}softmax(\frac{\mathbf{W}^\top \hat{z}^{i\to t}}{\tau})$ and $p^{t\to i} = softmax(\frac{\mathbf{W}^\top \hat{z}^{t\to i}}{\tau})$, respectively. Re-scaling the probabilities with temperature τ raises the output entropy so better image-text matching can be observed [189]. Subsequently, we use KL-divergence to measure the differences between the re-scaled probabilities. Since the magnitudes of the gradients produced by the re-scaling probabilities scale as $1/\tau^2$, it is important to multiply them by τ^2. Finally, the KL-divergence loss on the scaling probabilities for data imbalance can be formulated as L_{di}:

$$\begin{aligned} L_{di} = & \frac{\tau^2}{N} \sum \sum \Big(p^{i\to t} * log(\frac{p^{i\to t}}{p^{t\to i}+\varepsilon}) + p^{t\to i} * log(\frac{p^{t\to i}}{p^{i\to t}+\varepsilon}) \Big) \\ & s.t.\ \ p^{i\to t}{=}softmax\Big(\frac{\mathbf{W}^\top \hat{z}^{i\to t}}{\tau}\Big), p^{t\to i}{=}softmax\Big(\frac{\mathbf{W}^\top \hat{z}^{t\to i}}{\tau}\Big) \end{aligned} \tag{4.11}$$

where ε is a small constant to avoid division by zero. With $\tau = 1$, we recover the original KL-divergence. As reported in Table 4.5, we find that the parameter τ can affect the effectiveness of loss L_{di}. Minimizing loss L_{di} effectively reduces the influence of data imbalance issue and improves retrieval accuracy. The final objective function for label classification is $(L_{ce} + L_{di})$. The gradients calculated from loss $(L_{ce} + L_{di})$ are used to optimize the parameters $\boldsymbol{\theta}_{E_1}$, $\boldsymbol{\theta}_{E_2}$, and $\boldsymbol{\theta}_P$ in the generator and the label classifier, respectively.

4.4.4 Bi-directional triplet constraint

The triplet constraint is commonly used for feature learning. To achieve the baseline performance, we use this constraint from an inter-modality and an intra-modality perspective to strengthen the discrimination of cross-modal features.

Given cross-modal features Z^i and Z^t in the shared space, the cosine function is used to measure global similarity between feature vectors, *i.e.* $S_{jk} = (Z^i_j)^\top Z^t_k$. We adopt the hard sampling strategy to select three-tuples features from an inter-modality and an intra-modality viewpoint. Hence, the inter-modality and intra-modality triplet loss functions are formulated as:

$$L_{inter} = \frac{1}{N}\Big(\sum_{j,k^+,k^-}^{N} \max[0, m - S_{j,k^+} + S_{j,k^-}] + \sum_{k,j^+,j^-}^{N} \max[0, m - S_{k,j^+} + S_{k,j^-}] \Big) \tag{4.12}$$

Algorithm 1: Whole network training and optimization pseudocode

Input: mini-batch images X^i, text X^t, instance label Y, modality label (Y_c^i, Y_c^t), total training batch S, pre-trained parameters $\boldsymbol{\theta}_{E_1}$, update steps k
Output: the embedded cross-modal features Z^i and Z^t in Figure 4.1
Initialize hash functions: learning rate lr_1, lr_2, $\boldsymbol{\theta}_{E_2}$, $\boldsymbol{\theta}_P$, $\boldsymbol{\theta}_D$
For $n = 1$ to S
  For k steps
    cross-modal features embedding:
    $Z^i = E_1(X^i; \boldsymbol{\theta}_{E_1})$ *//Embed image features into the shared space*
    $Z^t = E_2(X^t; \boldsymbol{\theta}_{E_2})$ *//Embed text features into the shared space*
    loss computing and feature optimization:
    $L_{ce}, L_{di}, L_{tr}, L_{kl}$ calculation *//Eqs. 4.10, 4.11, 4.14, 4.9*
    $P_D^i = D(Z^i; \boldsymbol{\theta}_D)$ *//Discriminator D*
    $P_D^t = D(Z^t; \boldsymbol{\theta}_D)$
    L_s, L_c calculation *//Eqs. 4.4, 4.5*
    fix $\boldsymbol{\theta}_D$, update parameters $\boldsymbol{\theta}_{E_1}$, $\boldsymbol{\theta}_{E_2}$, $\boldsymbol{\theta}_P$:
    $\boldsymbol{\theta}_P \leftarrow \boldsymbol{\theta}_P - lr_2 \cdot \nabla_{\boldsymbol{\theta}_P}(L_{ce} + L_{di})$
    $\boldsymbol{\theta}_{E_1} \leftarrow \boldsymbol{\theta}_{E_1} - lr_1 \cdot \nabla_{\boldsymbol{\theta}_{E_1}}(L_{ce} + L_{di} + L_{tr} + L_{kl} + L_s)$
    $\boldsymbol{\theta}_{E_2} \leftarrow \boldsymbol{\theta}_{E_2} - lr_2 \cdot \nabla_{\boldsymbol{\theta}_{E_2}}(L_{ce} + L_{di} + L_{tr} + L_{kl} + L_s)$
  End for
  fixate $\boldsymbol{\theta}_P$, $\boldsymbol{\theta}_{E_1}$, $\boldsymbol{\theta}_{E_2}$, update parameters $\boldsymbol{\theta}_D$:
  $\boldsymbol{\theta}_D \leftarrow \boldsymbol{\theta}_D - lr_2 \cdot \nabla_{\boldsymbol{\theta}_D}(L_c)$
End for

$$L_{intra} = \frac{1}{N}\Big(\sum_{j,j^+,j^-}^{N} \max[0, m - S_{j,j^+} + S_{j,j^-}] + \sum_{k,k^+,k^-}^{N} \max[0, m - S_{k,k^+} + S_{k,k^-}] \Big) \tag{4.13}$$

$$L_{tr} = L_{inter} + L_{intra} \tag{4.14}$$

where m is the margin in the bi-directional triplet loss function. For instance, in case of inter-modality, $S_{j,k^+} = (Z_j^i)^\top Z_{k^+}^t$, where the anchor features are selected from the visual modality, while the positive features are selected from the textual modality. In case of intra-modality, $S_{j,j^+} = (Z_j^i)^\top Z_{j^+}^i$, both the anchor features and the positive features are selected from the visual modality. Minimizing bi-directional triplet loss L_{tr} keeps the correlated image-text pairs closer to each other, while the uncorrelated image-text pairs are pushed away. This loss directly operates on the cross-modal features Z^i and Z^t so that the gradients from it optimize the parameters $\boldsymbol{\theta}_{E_1}$ and $\boldsymbol{\theta}_{E_2}$ of the generator.

The problem of integrating information theory and adversarial learning for cross-modal retrieval is formally defined, in Eq. 4.15, as a min-max game using the previously defined loss terms. We further introduce the complete procedure of training and optimization in Algorithm 1. Finally, when trained to convergence, the network

yields cross-modal features Z^i and Z^t in the shared space, as shown in Figure 4.1. These return cross-modal features are used for performing retrieval.

$$\begin{cases} \min\limits_{\boldsymbol{\theta}_{E_1},\boldsymbol{\theta}_{E_2},\boldsymbol{\theta}_P} \max\limits_{\boldsymbol{\theta}_D}(L_{ce} + L_{di} + L_{kl} + L_{tr} + L_s) \\ \min\limits_{\boldsymbol{\theta}_D} L_c \end{cases} \tag{4.15}$$

4.5 Experiments

4.5.1 Datasets and settings

We demonstrate the efficacy of the proposed method on the Flickr8K [190], Flickr30K [191], Microsoft COCO [192], and CUHK-PEDES [193] datasets. Each image in these datasets is described by several descriptive sentences. For Flickr8K, we adopt the standard dataset splitting method to obtain a training set (6K), a validation set (1K), and a test set (1K). For Flickr30K, we follow the previous work [186] and use 29,783 images for training, 1,000 images for validation and 1,000 images for testing. For MS-COCO, we follow the training protocol in [186] and split this dataset into 82,783 training, 30,504 validation and 5,000 test images, and then report the performance on both 5K and 1K test set. For CUHK-PEDES, it contains 40,206 pedestrian images of 13,003 identities. Following [186], we split this dataset into 11,003 training identities with 34,054 images, 1,000 validation identities with 3,078 images and 1,000 test identities with 3,074 images. Note that all captions for the same image are used as separate image-text pairs to train network.

Models are trained on GEFORCE TITAN X and Tesla K40 GPUs. To extract text features, the embedded words are fed into a Bi-LSTM to capture vectors with dimension 1024 (1024-D). We follow [186] and set the Bi-LSTM with dropout rate 0.3. For fair comparison, we adopt ResNet [13], MobileNet [187], and VGGNet [61] as the backbone to extract image features and further fine-tune them with learning rate $lr_1 = 2 \times 10^{-5}$, decaying every 2 epochs exponentially. The output 2048-D image features and 1024-D text features are further projected into a shared space. Then cross-modal features in the space are 512-D vectors (*i.e.* Z^i and Z^t in Figure 4.1). The batch size is set to 64 or 32 depending on available GPUs memory. For the bi-directional triplet loss function, initially, we treat the inter-modality and intra-modality sampling identically although each of them might have different contributions [194], we empirically set the margin to $m = 0.5$. The re-scaling parameter τ for data imbalance issue is set as $\tau = 4$ (see Table 4.5). In practice, the discriminator can classify image and text modality easily at the start of training, so the generator typically requires multiple (*e.g.*, 5) update steps per discriminator update step during training (see Algorithm 1).

Once trained to converge, the network yields image features Z^i and text features Z^t. We use the cosine function to measure their similarity. We use Recall@K (K=1,

Table 4.1: Comparison of retrieval results on the Flickr30K [191] and MS-COCO [192] dataset (R@K (K=1,5,10)(%))

Method	Backbone Net	Flickr30K						MS-COCO					
		Image-to-Text			Text-to-Image			Image-to-Text			Text-to-Image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
m-RNN [196]	VGG	35.4	63.8	73.7	22.8	50.7	63.1	41.0	73.0	83.5	29.0	42.2	77.0
RNN+FV [197]	VGG	35.6	62.5	74.2	27.4	55.9	70.0	41.5	72.0	82.9	29.2	64.7	80.4
DSPE+FV [194]	VGG	40.3	68.9	79.9	29.7	60.1	72.1	50.1	79.7	89.2	39.6	75.2	86.9
CMPM+CMPC† [186]	MobileNet	40.3	66.9	76.7	30.4	58.2	68.5	52.9	83.8	92.1	41.3	74.6	85.9
Word2VisualVec [198]	ResNet-152	42.0	70.4	80.1	-	-	-	-	-	-	-	-	-
sm-LSTM [199]	VGG	42.5	71.9	81.5	30.2	60.4	72.3	53.2	83.1	91.5	40.7	75.8	87.4
RRF-Net [200]	ResNet-152	47.6	77.4	87.1	35.4	68.3	79.9	56.4	85.3	91.5	43.9	78.1	88.6
Joint learning [143]	ResNet-152	48.6	73.6	83.6	32.3	62.5	74.0	55.3	82.7	90.2	41.7	75.0	87.4
CMPM+CMPC‡ [186]	ResNet-152	49.6	76.8	86.1	37.3	65.7	75.5	-	-	-	-	-	-
VSE++ [184]	ResNet-152	52.9	80.5	87.2	39.6	70.1	79.5	51.3	82.2	91.0	40.1	75.3	86.1
TIMAM [201]	ResNet-152	53.1	78.8	87.6	42.6	71.6	81.9	-	-	-	-	-	-
DAN [202]	ResNet-152	55.0	81.8	89.0	39.4	69.2	79.1	-	-	-	-	-	-
Dual-path stage I [203]	ResNet-152	44.2	70.2	79.7	30.7	59.2	70.8	52.2	80.4	88.7	37.2	69.5	80.6
Dual-path stage II [203]	ResNet-152	55.6	81.9	89.5	39.1	69.2	**80.9**	**65.6**	**89.8**	**95.5**	47.1	79.9	90.0
Our ITMeetsAL	VGG	38.5	66.5	76.3	30.7	59.4	70.3	44.2	76.1	86.3	37.1	72.7	85.1
Our ITMeetsAL	MobileNet	46.6	73.5	82.5	34.4	63.3	74.2	54.7	84.3	91.1	41.0	76.7	88.1
Our ITMeetsAL	ResNet-152	**56.5**	**82.2**	**89.6**	**43.5**	**71.8**	80.2	58.5	85.3	92.1	**48.3**	**82.0**	**90.6**

MS-COCO is tested on 1K setting. The best results are in boldface and the second best ones are underlined.

5, 10) for evaluation and comparison. Moreover, we adopt the precision-recall and mAP for the ablation study, and visualize their feature distributions by t-SNE [195]. Furthermore, we display the cross-modal retrieval results using our method.

4.5.2 Performance evaluation

4.5.2.1 Results on the Flickr30K and MS-COCO datasets

The retrieval results on Flickr30K and MS-COCO are reported in Table 4.1. Hereafter, "Image-to-Text" means using an image as a query item to retrieve semantically-relevant text from the textual gallery. "Text-to-Image" means using a text as query to retrieve images from the visual gallery. In most cases, our proposed approach shows the best performance when using three different deep networks. For the "Image-to-Text" task on the MS-COCO dataset, the best results are obtained by Zheng *et al.* [203], which adopted a deeper network for text feature learning and used a two-stage training strategy. However, for the "Text-to-Image" task and the "Image-to-Text" task on the Flickr30K dataset, our method performs better. Take ResNet-152 as an example, the results are R@1=43.5% on the Flickr30K and R@1=48.3% on the MS-COCO for "Text-to-Image" task; the results are R@1=56.5% on the Flickr30K dataset and R@1=58.5% on the MS-COCO dataset for "Image-to-Text" task.

The learning capacity of deep networks would affect retrieval performance significantly. For visual feature learning, deeper CNNs usually achieve better results than their shallower counterparts. This can be observed from Table 4.1, the retrieval results based on ResNet-152 are usually higher than those of MobileNet and VGG. Moreover, our method also has good performance using MobileNet. For instance, regarding the "Image-to-Text" task on the Flickr30K dataset, the recall

Table 4.2: Retrieval results on the CUHK-PEDES [193] dataset.

Method	Backbone Net	Text-to-Image		
		R@1	R@5	R@10
Latent co-attention [204]	VGG	25.94	-	60.48
Local-global association [205]	ResNet-50	43.58	66.93	76.26
CMPM [186]	MobileNet	44.02	-	77.00
Dual-path two-stage [203]	ResNet-152	44.40	66.26	75.07
MIA [206]	ResNet-50	48.00	70.70	79.30
CMPM+CMPC [186]	MobileNet	49.37	-	79.27
Our ITMeetsAL	VGG	44.43	68.26	77.50
Our ITMeetsAL	MobileNet	51.85	73.36	81.27
Our ITMeetsAL	ResNet-50	50.63	73.33	81.34
Our ITMeetsAL	ResNet-152	**55.72**	**76.15**	**84.26**

result of CMPM+CMPC [186] is R@1=40.3%, but the result from our method is R@1=46.6%, which is a significant improvement. Likewise, for textual modality, a powerful extractor provides better semantic-aware features, providing better results. This can be observed on the comparisons between our proposed "ITMeetsAL", m-RNN [196] and RNN+FV [197]. Concretely, both of them leverage VGG to extract image features, but m-RNN [196] and RNN+FV [197] extract textual features using RNN, which is less powerful than the Bi-LSTM as we used in our experiments.

We obverse that the strategy for network training is critical for retrieval tasks. Take [203] as an example, the backbone network (ResNet-152) is fixed at stage I (R@1=44.2% on "Image-to-Text" task on Flickr30K) and then fine-tuned with a small learning rate on stage II (R@1=55.6% on the "Image-to-Text" task on Flickr30K). In contrast, our network structure is trained end-to-end in only one stage (we fine-tune the backbone network with a small learning rate from the beginning). Our reported results are close to those in two-stage dual learning [203]. When tested on the Flickr30K dataset for the "Image-to-Text" task, the recall results are R@1=56.5%, R@5=82.2%, R@10=89.6%, which are the best overall previous methods.

Considering the two branches of "Image-to-Text" task and the "Text-to-Image" task, we think that the data imbalance issue still influences the performance of each branch. More specifically, for all listed methods, the "Image-to-Text" task has better performance, which indicates that the network still has more biases on text feature learning as a result of the issue of data imbalance. Thus, there exists more room for improvement using other strategies, such as data augmentation.

4.5.2.2 Results on CUHK-PEDES dataset

The "Text-to-Image" retrieval results on the CUHK-PEDES dataset are reported in Table 4.2. We evaluate the proposed method using four deep networks. All results indicate that our method outperforms other counterparts. The optimal results are achieved with R@1=55.72% using ResNet-152 as backbone network. The results using MobileNet are sub-optimal but also have some improvements. For

Table 4.3: Retrieval results on the Flickr8K [190] dataset (R@K (K=1,5,10)(%))

Method	Backbone Net	Image-to-Text		
		R@1	R@5	R@10
RNN+FV [197]	VGG	23.2	53.3	67.8
GMM+HGLMM [207]	VGG	31.0	59.3	73.7
Word2VisualVec [198]	ResNet-152	33.4	63.1	75.3
Joint learning [143]	ResNet-152	**40.6**	**67.8**	78.6
Our ITMeetsAL	VGG	28.0	52.7	63.1
Our ITMeetsAL	MobileNet	30.9	58.6	70.8
Our ITMeetsAL	ResNet-152	40.1	**67.8**	**79.2**

The best results are in boldface and the second best results are underlined.

Table 4.4: Component analysis on the Flickr30K [191] (R@1, R@10, and mAP (%))

Method using MobileNet	Flickr30K					
	Image-to-Text			Text-to-Image		
	R@1	R@10	mAP	R@1	R@10	mAP
Baseline1: Only L_{ce}+L_{tr}	40.6	80.8	23.1	31.9	72.2	31.9
Baseline2: L_{ce}+L_{tr}+L_{di}	42.3	80.6	24.4	32.5	73.0	32.5
Baseline3: L_{ce}+L_{tr}+L_{di}+L_{kl}	44.7	81.0	25.2	32.6	73.2	32.6
Full method: L_{ce}+L_{tr}+L_{di}+L_{kl}+L_s+L_c	46.6	82.5	26.3	34.4	74.1	34.4

example, CMPM+CMPC achieves a recall R@1=49.37% and R@10=79.27%, while our method obtains R@1=51.85% and R@10=81.27%. Moreover, the results of our method show that deeper networks achieve better retrieval performance, whereas the light-weight MobileNet has a similar performance as ResNet-50.

4.5.2.3 Results on Flickr8K dataset

The retrieval results on the Flick8K dataset are reported in Table 4.3. The best results R@1=40.6%, R@5=67.8%, R@10=78.6% are achieved by joint correlation learning [143] where a batch-based triplet loss, which considers all image-sentences pairs, is used for learning correlations. The second-best results are achieved using ResNet-152 (same as [143]) R@1=40.1%, R@5=67.8%, R@10=79.2%, which has better R@10 performance compared to [143]. Our method shows competitive results compared to other counterparts and also indicates that there exists room for further performance improvement.

4.5.3 Ablation study

For analyzing the effect of each component, the ablation study are conducted on the Flickr30K dataset using MobileNet as a backbone net, we use the commonly used categorical cross-entropy L_{ce} and bi-triplet loss function L_{tr} to construct the baseline in Table 4.4, we call this **Baseline1** configuration "Only $L_{ce} + L_{tr}$".

4.5.3.1 Analysis of KL-divergence for data imbalance

Each image in a dataset (*e.g.* Flickr30k) has more than one description sentence. We think this leads to a data imbalance issue for cross-modal feature learning. The network has more text data for training, which causes the learned label classifier to prefer text features. Therefore, we adopt a regularization term L_{di} based on KL-divergence to calibrate this bias. To this end, the label classifier can be re-calibrated on the image features and text features. In Table 4.4, this **Baseline2** configuration is named " $L_{ce} + L_{tr} + L_{di}$". The Recall and mean Average Precision (mAP) show the effectiveness of this loss. Compared to Baseline1, the scaling KL-divergence loss L_{di} contributes more on Recall@1 for both the "Image-to-Text" (42.3%) and "Text-to-Image" task (32.5%).

4.5.3.2 Analysis of KL-divergence for cross-modal feature projection

KL-divergence is obtained by adding L_{kl} which constrains the image features and text features in the shared space under the supervision of supervisory matrix. It focuses on the whole feature distribution and is complementary to the bi-directional triplet loss function. We denote **Baseline3** as "$L_{ce}+L_{tr}+L_{di}+L_{kl}$" in Table 4.4. As we can see, Recall@1 of the "Image-to-Text" task has been improved significantly by 2.4%. However, the KL-divergence loss shows a slight improvement on the "Text-to-Image" task. The results indicate that the KL-divergence loss function contributes more to image feature learning, which might be caused by the issue of data imbalance of the dataset.

4.5.3.3 Analysis of adversary combining

The prior loss terms have been used to constrain the similarity of the image-text features in the shared space. Intuitively, two-tuple or three-tuple feature exemplars are helpful for reducing the "semantic gap" and further making the whole feature distribution close at the same time. However, the constraint loss functions (*e.g.* cosine similarity) cannot constrain the distribution discrepancy of the whole distribution because these loss functions are symmetrical. Focusing on the whole feature distribution, we combine the Shanon information entropy L_s and the modality classification loss L_c in an adversary training manner to reduce the heterogeneity gap. This **full method** is named "$L_{ce} + L_{tr} + L_{di} + L_{kl} + L_s + L_c$" and corresponding results are shown in Table 4.4. Compared to former baselines, the results obtained by using our method are improved significantly.

Furthermore, we compare the precision-recall curves for the above four configurations and baselines, the results are shown in Figure 4.4. The larger the area under the curve, the better the algorithm. Regarding the different tasks, the improvements are slightly different. Overall, we can see that each added component helps to improve the overall performance of the retrieval algorithm.

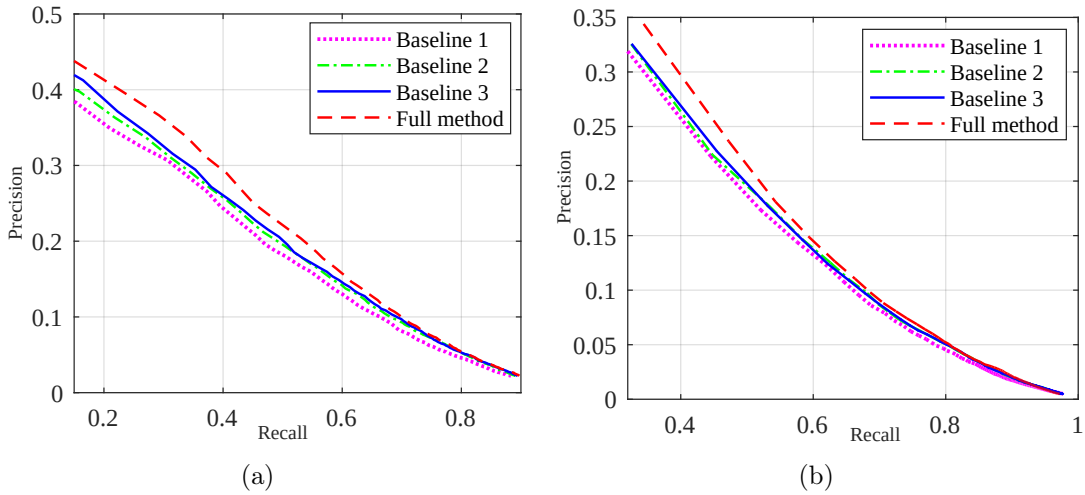


Figure 4.4: The precision_recall curves from "Baseline1" to "Full method" on Flickr30K, each line corresponds one experimental configuration in Table 4.4. The larger area under the line indicates better performance.

Table 4.5: Temperature scaling analysis for loss L_{di} (R@1, R@10, and mAP (%))

Temperature	Flickr30K					
	Image-to-Text			Text-to-Image		
	R@1	R@10	mAP	R@1	R@10	mAP
τ=1	44.0	80.6	24.8	32.9	73.5	32.9
τ=2	45.3	80.9	25.6	33.6	73.6	33.6
τ=3	46.2	**83.2**	25.7	33.3	73.4	33.3
τ=4	**46.6**	82.5	**26.3**	**34.4**	**74.2**	**34.4**
τ=5	46.0	81.6	26.1	34.3	73.9	34.3
τ=6	45.9	80.2	26.1	33.1	73.4	33.1

4.5.3.4 Analysis of temperature τ

We analyze the temperature parameter τ in loss L_{di} in Eq. 4.11. Other loss terms are kept the same with the full method, *i.e.* "$L_{ce} + L_{tr} + L_{di} + L_{kl} + L_s + L_c$". We vary this parameter τ from 1 to 6, and their corresponding results are reported in Table 4.5. We can observe that the optimal results are achieved if the classifier's output probabilities are re-scaled by $\tau = 4$. As claimed in [189], the temperature scaling raises the output entropy of the classifier with $\tau > 1$. In our experiments, we found it is beneficial for improving the image-text matching.

4.5.3.5 Distribution visualization

We choose 40 image-text pairs from the Flickr30K dataset to visualize their feature distributions using t-SNE [195]. We only choose the first description caption among the five sentences. In Figure 4.5, the circle and the triangle shape denote text features and image features, respectively. Label information is represented by a different color.

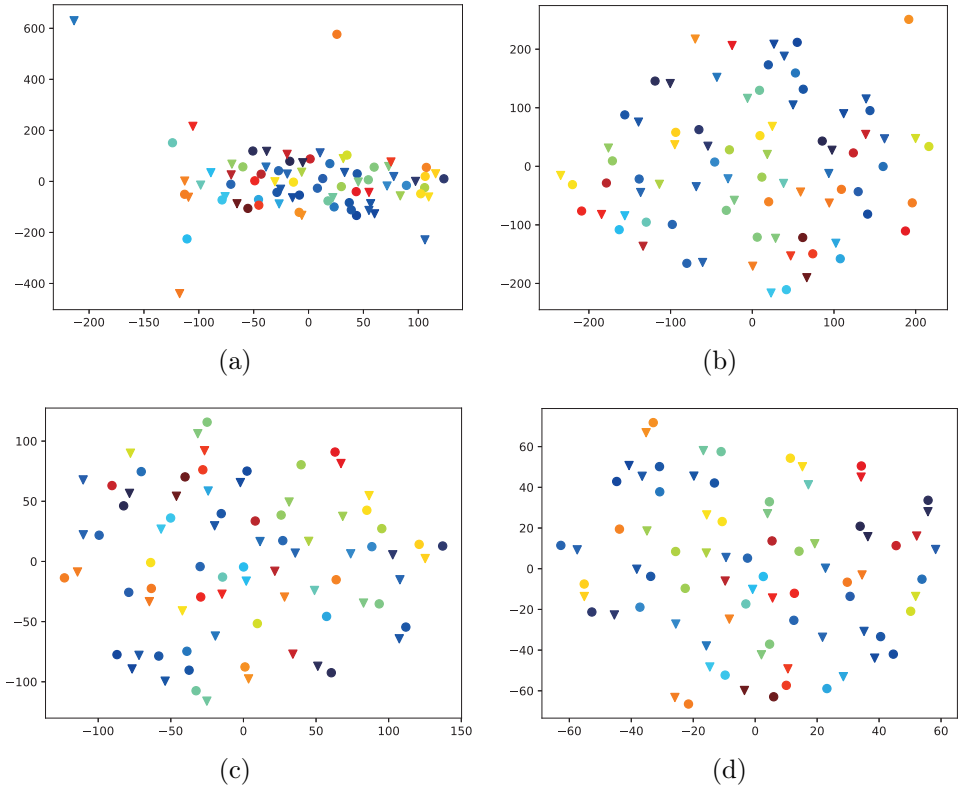


Figure 4.5: Feature distribution visualizations for the ablation study. The shape represents modality and the color indicates the label information. Sub-figures (a)∼(d) correspond to the four experimental configurations in Table 4.4. When each loss function is gradually applied, the paired image features and text features have smaller distances. Best viewed in color.

This distribution indicates the effectiveness of each component (*e.g.* KL-divergence for cross-modal feature projection, and the Shannon information entropy trained in an adversarial manner). In Figure 4.5(a), there exist several feature outliers within the distribution and the proximity relationship between pair-wise features is not obvious. When using the proposed components, the features distribute much better. For example, in Figure 4.5(d), all loss functions are utilized to constrain feature learning, the pair-wise feature shows a close proximity relationship. Moreover, image features and text features are distributed within smaller ranges (-60 $\sim$ 60). Few outliers exist among the whole distribution.

Qualitative retrieval results on the Flickr30K and the CUHK-PEDES dataset are shown in Figure 4.9. For the "Image-to-Text" task, the proposed method can return almost all paired text of the query image. The "Image-to-Text" task also has good performance, the proposed method retrieves the paired image correctly. Also, other retrieved images show contents relevant to the query sentence.

4.5.3.6 Analysis of complexity and stability

We analyze the complexity of the proposed method by evaluating FLOPs (network forward pass), parameter size, and inference time for each image-text pair on Flickr8k. The results are reported in Table 4.6. The complexity of the proposed framework, implemented by three networks, performs differently. It is well known that VGG has a larger model size and more parameters, which increase computation cost. As a result, the FLOPs of the VGG-based framework achieve 3.1×10^{10}, while the lightweight MobileNet reaches FLOPs to 1.1×10^{9}. Although ResNet-152 has more layers than VGG and MobileNet, it achieves in-between FLOPs to 2.2×10^{10}. The model complexity also leads to different inference times for each image-text input. Take MobileNet on Flickr30k as an example, its inference time is 14.8±3.2 *ms*, relatively faster than these of VGG and ResNet-152.

Table 4.6: Comparisons of model size and computation complexity. FLOPs: the number of FLoating-point OPerations;

Dataset	Backbone Net	FLOPs (forward pass)	#Parameters (million)	Inference time (ms) (per image-text pair)
Flickr8k	Based on VGG	3.1×10^{10}	147.2	114.32±2.5
	Based on MobileNet	1.1×10^{9}	14.6	15.6±3.1
	Based on ResNet-152	2.2×10^{10}	70.1	110.6±2.3

An algorithm is stable if it produces consistent predictions with respect to small perturbations of training samples [208, 209, 210]. Therefore, stability of a learning algorithm holds if statistical conclusions are robust or stable to appropriate perturbations to data [209]. According to this definition, we conduct a stability analysis based on Flickr8k using MobileNet. We add Gaussian noise $N \sim \mathcal{N}(\mu, \sigma^2)$ to change the image-text pairs, with a varying σ. For this purpose, first, we build up an upper-bound performance where no Gaussian noise is added. Second, we vary the σ and collect the corresponding output and then evaluate its Recall rate.

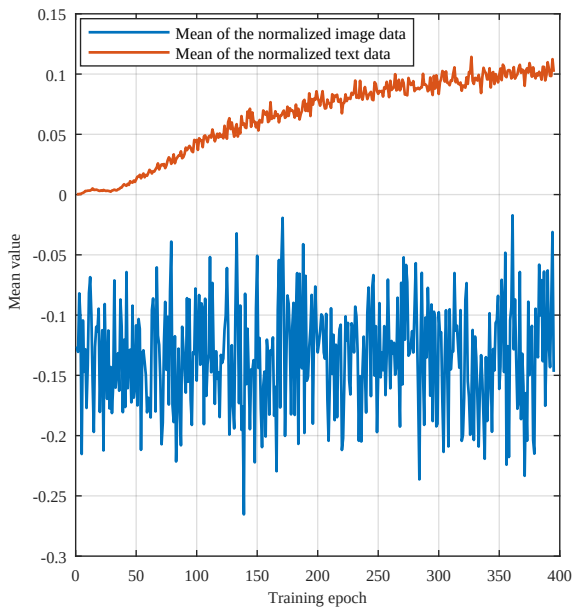
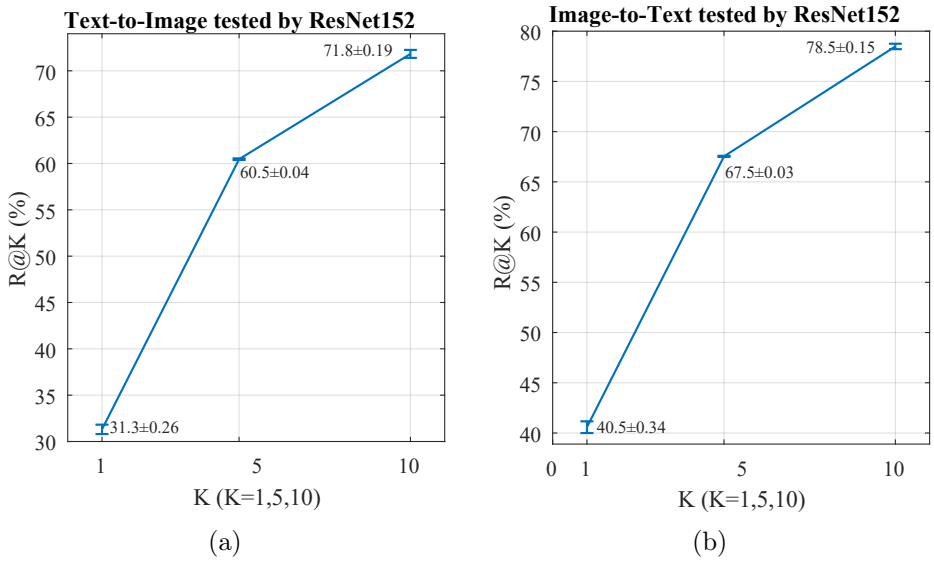


Figure 4.6: The means of the input image and text data (after normalization).

Since the training data have been normalized before feeding into the network and they have small means, as depicted in Figure 4.6. Since the magnitude of image and text inputs are small, we determine the mean of the Gaussian noise by the corresponding means of image and text inputs in each training epoch.

Table 4.7: Stability evaluation on Flickr8k using MobileNet as a backbone net.

Gaussian distribution		Text-to-Image			Image-to-Text		
		R@1	R@5	R@10	R@1	R@5	R@10
With no noise added		23.8	49.7	61.3	30.8	58.9	70.3
$\mu_x = Mean(X^i)$ $\mu_y = Mean(X^t)$	$\sigma = 0.025$	23.8	50.3	61.3	30.7	58.9	71.2
	$\sigma = 0.05$	22.7	47.5	59.0	30.3	56.6	68.6
	$\sigma = 0.075$	22.7	48.0	59.2	29.8	54.8	66.7
	$\sigma = 0.1$	22.2	46.6	58.3	27.7	54.0	65.3



(a) (b)

Figure 4.7: Error analysis for the proposed model on Flickr8k based on ResNet-152.

The averaged results are reported in Table 4.7. We vary the variance from 0.025 to 0.1, the performance of the proposed framework is relatively stable. For example, for the "Text-to-Image" task, when varying $\sigma = 0.025$ to $\sigma = 0.1$, the result of R@1 changes from 23.8% to 22.2%, decreasing by about 6.7%.

Besides, we also perform error analysis for the performed framework on Flickr8k using ResNet-152. For "Text-to-Image" and "Image-to-Text" tasks, we consider the error bar calculation based on three times running. The results of R@K (K=1,5,10) are illustrated in Figure 4.7. In this error analysis, we observe that the recall results for "Text-to-Image" and "Image-to-Text" tasks have small variations.

4.5.4 Further exploring

We propose to integrate Shannon information entropy with the discriminator for cross-modal retrieval. That is, the discriminator performs modality classification and measures the information entropy at the same time (see Figure 4.3). Herein, we further explore a paradigm to integrate information entropy with adversarial learning. This combining paradigm is more straightforward to the structure in Figure 4.1. Concretely, we build two branches of sub-networks: an uncertainty predictor for

modality uncertainty prediction and a modality classifier for modality classification. Then adversarial learning is implemented as an interplay between these two subnetworks with competitive objectives. The uncertainty predictor aims at maximizing the modality uncertainty of the shared space (measured by information entropy), while the modality classifier is to identify image inputs and text inputs by modality classification. We illustrate this combining paradigm in Figure 4.8. Compared to the former paradigm depicted in Figure 4.3, the optimization depicted in Figure 4.8 is different and more complex. The gradients computed by the classifier are used to update parameters $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_T$ in the feature extractor. To learn modality-invariant features, the feature extractor *minimizes* the loss of the uncertainty predictor and it *maximizes* the loss $L_d = L_c$ (Eq. 4.5) of the modality classifier, which aims to make image features and text features as similar as possible [211]. The parameters of the modality classifier *minimize* its loss L_d. This training process needs to depend on the gradient reversal layer [211], which would multiply gradient values by -1 when executing back-propagating.

The training procedure is almost the same as used in Algorithm 1 except for the gradients from the modality classification loss that updates the backbone network, leading to a slower training process. The retrieval performance of these two combined methods presented in Figure 4.3 and Figure 4.8 (named as unified and separate, respectively) are given in Table 4.8. The backbone net for image feature extraction is ResNet-152. These two combined strategies show different performances on the four datasets when combining information entropy and modality classification into a unified discriminator. The performance improves slightly on the Flickr30K, MS-COCO, and Flickr8K datasets when adopting the combining strategy of Figure 4.3. However, the method depicted in Figure 4.8 has better performance on the CUHK-PEDES dataset, which is not the common objects dataset. This method has R@1 improved by 3.3% (from 65.58% to 67.79%), Also, the mAP has improved by 1.8% compared to the unified method depicted in Figure 4.3. In summary, the proposed framework of combining information entropy and adversarial learning in Figure 4.3 has better performance and has faster convergence during training.

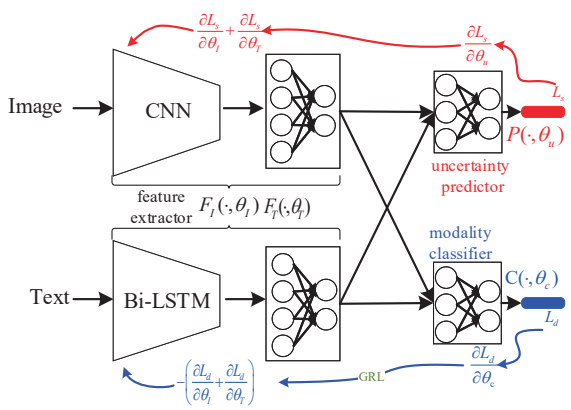


Figure 4.8: The illustration of independent combining information entropy and modality classification into an adversary, which is an intuitive structure of the diagram in Figure 4.1. Other loss functions are kept the same, but we do not show in this graph for simplicity. The gradients computed from the modality classifier in this combining paradigm are used to optimize the parameters $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_T$ of the feature extractor.

Table 4.8: Comparison of two combining paradigms on four retrieval datasets (R@1, R@10, and mAP(%))

Image-to-Text													
Combining strategy	Backbone Net	Flickr30K			MS-COCO			CUHK-PEDES			Flickr8K		
		R@1	R@10	mAP	R@1	R@10	mAP	R@1	R@10	mAP	R@1	R@10	mAP
Method in Figure 4.8	ResNet-152	55.30	88.30	32.23	57.00	92.10	35.12	67.79	93.75	34.79	39.00	77.70	22.33
Method in Figure 4.3	ResNet-152	56.50	89.60	32.58	58.50	92.10	36.28	65.58	93.60	34.17	39.90	77.90	22.46

4.6 Chapter Conclusions

In this work, we explored methods to improve the performance of cross-modal retrieval by integrating information theory and adversarial learning by analyzing the relation between information entropy and modality uncertainty. Based on this relation, we explored two different paradigms to combine information entropy maximization and modality classification in an adversarial manner. Training these two components iteratively reduces feature distribution discrepancies and further the heterogeneity gap. This is beneficial for preserving semantic similarity between cross-modal features by using bi-directional triplet loss and cross-entropy loss. In addition, we also considered the issue of data imbalance, which leads to a biased classifier and affects label classification. KL-divergence is used as an additional loss term to regularize the re-scaled probabilities computed from image features and text features. It is also used to constrain the cross-modal feature projections and is helpful for learning modality-invariant features. The efficacy of the proposed method was demonstrated by thorough experimental results on four well-known datasets using four deep models.

Successfully combining information entropy and adversarial learning depends on the competitive goals between the information entropy predictor and the modality classifier, and this leads to challenging directions worth further investigation. For example, we used instance labels as supervisory information in this work. Then the information entropy loss was computed only based on image modality and text modality. However, retrieval performance depends on the matching of each image-text feature pair. For some large-scale datasets, each category may include a large number of image-text pairs. Thus, it is valuable to make the information entropy loss specific for each category so that the discrepancy between two modalities can be reduced more granularly. Moreover, the problem of data imbalance leads to training a biased label classifier, which is an issue that can also be resolved by training strategies like data augmentation or by using other loss functions, *e.g.* knowledge distillation loss.

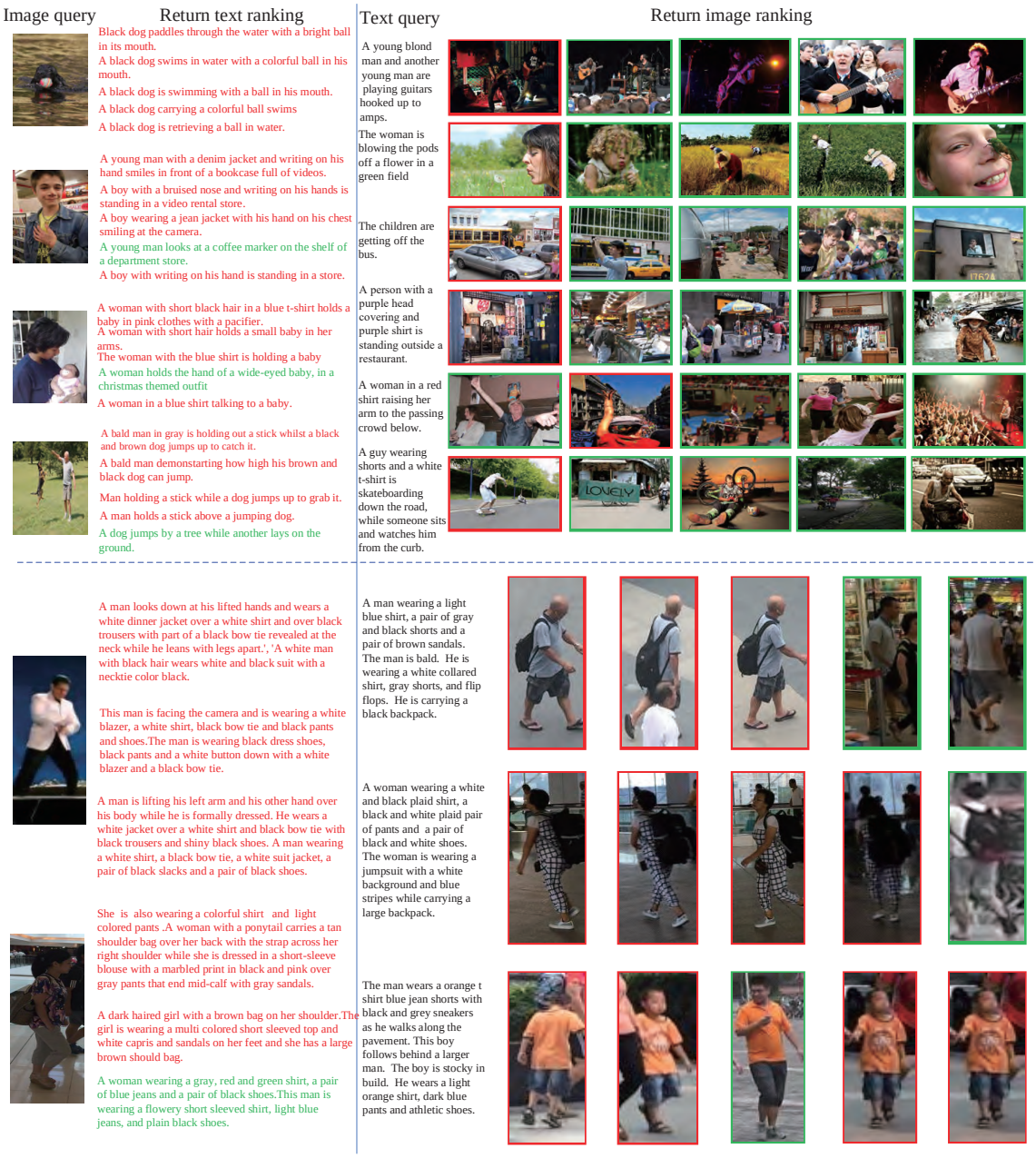


Figure 4.9: Qualitative test results on the Flikcr30K and CUHK-PEDES datasets. We report Recall@5 of the "Image-to-Text" task and the "Text-to-Image" task from left to right. The correct retrieval images or text are in red and a red box, while the failure retrieval are in green. For Flickr30K, each image is described by 5 sentences. Hence, each text query also has a correct retrieved image, but other retrieved images have similar content as described by the sentence. For the CUHK-PEDES dataset, each category has more than one image, thus almost all correct images are retrieved according to the text query. The list is best viewed in color.

Chapter 5

On the Exploration of Incremental Learning for Fine-grained Image Retrieval

As noted, the wide popularity of mobile devices make the large image collections available to access. Deep models are usually trained for retrieval on limited categories and cannot be extended to new incoming data. To satisfy a more practical retrieval, deep models are required to learn on a stream of data sequentially. This motivates our research on what kind of knowledge is more beneficial for making a deep model learn incrementally and reduce catastrophic forgetting.

In this chapter, we consider the problem of fine-grained image retrieval in an incremental setting, when new categories are added over time. On the one hand, repeatedly training the representation on the extended dataset is time-consuming. On the other hand, fine-tuning the learned representation only with the new classes leads to catastrophic forgetting. To this end, we propose an incremental learning method to mitigate retrieval performance degradation. Without accessing any samples of the original classes, the classifier of the original network provides soft "labels" to transfer knowledge to train the adaptive network, so as to preserve the previous capability for classification. More importantly, a regularization function based on Maximum Mean Discrepancy is devised to minimize the discrepancy of new classes features from the original network and the adaptive network, respectively.

Keywords

Incremental learning, Fine-grained image retrieval, Knowledge distillation, Feature correlation, Maximum mean discrepancy

This chapter is based on the following publication [37]:

- Chen, W., Liu, Y., Wang, W., Tuytelaars, T., Bakker, E, and Lew, M.S., "On the Exploration of Incremental Learning for Fine-grained Image Retrieval." The British Machine Vision Conference (BMVC), 2020, pp. 1-10.

5.1 Introduction

In an era when the number of images is increasing, deep models for fine-grained image retrieval (FGIR) are required to be adaptable for new incoming classes. However, current image retrieval approaches are focusing mainly on static datasets and are not suited for incremental learning scenarios. To be specific, deep networks well-trained on original classes will under-perform on new incoming classes.

When new classes are added into an existing dataset, joint training on all classes allows to guarantee the performance. However, as the number of new classes increases sequentially, the repetitive re-training is time-consuming. Alternatively, fine-tuning makes the network adapt to new classes and achieve good performance on these classes. However, when the original classes become inaccessible during fine-tuning, the performance of the original classes degrades dramatically because of catastrophic forgetting, a phenomenon that occurs when a network is trained sequentially on a series of new tasks and the learning of these tasks interferes with performance on previous tasks, as shown in Figure 5.1(a).

Most of incremental learning methods are exploited for image classification, which is robust and forgiving as long as features remain within the classification boundaries. In contrast, image retrieval focuses more on the discrimination in the feature space rather than the classification decisions. Especially for FGIR, small changes on visual features may have a big impact on the retrieval performance. Additionally, we find that standard methods like Learning without Forgetting (*i.e.* LwF [212]) and Elastic Weight Consolidation (*i.e.* EWC [213]) are insufficient for this problem because the distillation is not on the actual feature space (see Section 5.4.2 and 5.4.3).

Considering the above limitations, we alleviate the problem of incremental fine-grained image retrieval. We regularize the updates of the model to simultaneously retain preservation on original classes and adaptation on new classes. Importantly, to avoid the repeated training, the samples of the original classes are not used when learning the new classes. The classifier of the original network provides soft "labels" to transfer knowledge to train the adaptive network using the distillation loss function [214]. This focuses on pair-wise similarity but can not well quantify the distance between two feature distributions. This limitation inspires us to adopt a regularization term based on Maximum Mean Discrepancy (MMD) [215] to minimize the discrepancy between the features derived from an original network and an adaptive network, respectively. Moreover, the cross-entropy loss and triplet loss are utilized to identify subtle differences among sub-categories.

The novelty of the proposed method can be summarized two-fold. First, our work extends FGIR in the context of incremental learning. This is the first work to study this problem, to the best of our knowledge. Second, we propose a deep network, which includes a knowledge distillation loss and a MMD loss, for incremental

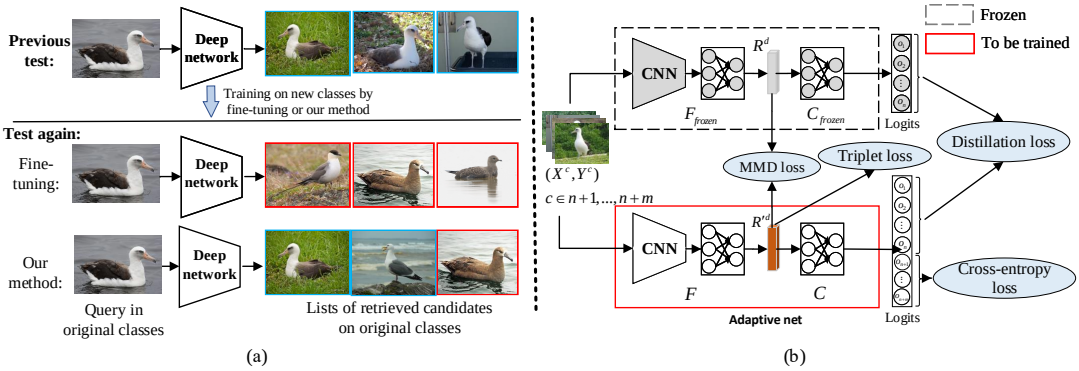


Figure 5.1: (a) Illustration of catastrophic forgetting for FGIR. Our method aims to maintain good performance on the original classes where the inaccurate returned images are in red box and correct results are in blue box. (b) Framework of our method. The only inputs for the adaptive net $\mathcal{B}$ are m new classes and labels $(\mathbf{X}^{c'}, \mathbf{Y}^{c'})$, $c' \in (n+1, ..., n+m)$. The frozen net $\mathcal{A}$ is firstly trained on n original classes and then copied as initialization for net $\mathcal{B}$.

learning without using any samples from the original classes. It achieves significant improvements over previous incremental learning methods.

5.2 Related Work

Incremental learning is the process of transferring learned knowledge from an original model to an incremental model. It has been studied in lots of tasks like image classification [212], image generation [216], object detection [217], hashing image retrieval [218], and semantic segmentation [219]. To overcome catastrophic forgetting, numerous methods have been proposed. For example, a subset of data (exemplars) of original classes are stored into an external memory, and the forgetting is thereby avoided by replaying these exemplars [220]. Recently, GANs [180] are used to synthesize samples with respect to the previous data distributions [221], which avoids the shortcomings of memory-consuming and exemplar-choosing, but generating real-like images with complex semantics is a challenging task. Alternatively, regularization methods constrain the objective functions or parameters of deep networks to preserve the previously learned knowledge. The distillation loss function [214] is used to transfer knowledge of old classes [212]. The importance weight per parameter is estimated based on the old classes, and then is used as regularization to penalize essential parameter changes when training on new incoming classes [213].

5.3 Proposed Approach

Problem formulation Given a fine-grained dataset which includes n class labels $(\mathbf{X}^c, \mathbf{Y}^c)$ where $c \in (1, ..., n)$, each sub-category c has a different amount of images in $\mathbf{X}^c$ and the ground-truth labels $\mathbf{Y}^c$. A deep network is trained to perform the

retrieval task for the n classes. Consider the incremental learning scenario, images from m new classes are added sequentially or at once. We take as input only the images from m new incoming classes, *i.e.* $(\mathbf{X}^{c'}, \mathbf{Y}^{c'})$, where $c' \in (n+1, ..., n+m)$, to incrementally train the deep network. In this way, it is efficient to update the network with no need of re-training the original classes again. Besides, the image instances from the original classes are not always accessible due to privacy issue or memory limit. Finally, the aim is to continually train the network, to make it preserve promising performance for all seen classes.

Overall idea As shown in Figure 5.1(b), our method includes two training stages. First, we train a network $\mathcal{A}$ on the original classes using cross-entropy and triplet loss on the output logits and representations. After $\mathcal{A}$ is well-trained, we make two copies of $\mathcal{A}$: one freezing its parameters when incrementally training, and the other adapting its parameters for the m incremental classes. We refer to this adaptive network as $\mathcal{B}$. It is initialized with parameters from $\mathcal{A}$, including the feature extraction layers F_{frozen} and classifier C_{frozen}, but extends the number of neurons in its classifier C, from which the output logits are $(o'_1, o'_2, \ldots, o'_n, o'_{n+1}, \ldots, o'_{n+m})$, and previous n neurons are copied from C_{frozen}. To overcome catastrophic forgetting, we propose to integrate two regularization strategies based on knowledge distillation and maximum mean discrepancy, respectively. Given a query image from either the original classes or newly added classes, we extract the features from the fully-connected layer for image retrieval. We introduce the details of our method below.

5.3.1 Semantic preserving loss

First, we train the model with the standard cross-entropy loss. Given the logits $(o_1, o_2, ..., o_n)$ and its class label $(y_1, y_2, \ldots, y_n)$, the loss is $H(\mathbf{y}, \mathbf{o}) = -\sum(\mathbf{y} * log(softmax(\mathbf{o})))$. Note that we only use images from the new classes during incremental training, thus the classification is performed on $(o'_{n+1}, o'_{n+2}, \ldots, o'_{n+m})$, the categorical cross-entropy loss function L_{ce} is

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i * log\big(\frac{e^{o'_i(x)}}{\sum_{j=n+1}^{n+m} e^{o'_j(x)}}\big)\Big) \tag{5.1}$$

To identify subtle differences among categories, we use the triplet loss $L_{triplet}$ by mining training samples based on feature vectors $\mathbf{R}$.

$$L_{triplet} = \frac{1}{N}\sum_{i=1}^{N}\Big(\max(0, \lambda + S_{i,neg} - S_{i,pos})\Big) \tag{5.2}$$

where $S_{i,neg}$ and $S_{i,pos}$, based on matrix multiplication (*i.e.* $S_{i,neg} = R_i R_{neg}^{\top}$), indicate the similarity of ith negative and positive pairs, respectively. λ is the margin.

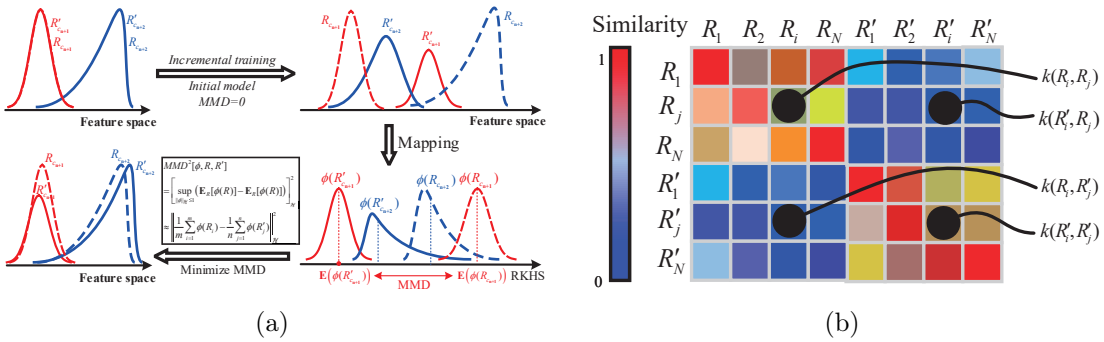


Figure 5.2: (a) The red and blue color depict the feature distributions of two categories. The dashed line indicates the distributions from the network $\mathcal{A}$, the solid line indicates that from the network $\mathcal{B}$. Since $\mathcal{A}$ is used to initialize the network $\mathcal{B}$, MMD=0 in the beginning. As training proceeds, $\mathcal{B}$ changes its output features and the MMD is expected to increase. (b) MMD for instance-to-instance similarity.

5.3.2 Knowledge distillation loss

We rewrite (F_{frozen}, C_{frozen}) as (F_f, C_f) for simplicity. Knowledge distillation loss [214] is defined to regularize the activations of the output layer in both the old and new model. To be specific, we constrain the first n values in $(o'_1, o'_2, ..., o'_n, o'_{n+1}, ..., o'_{n+m})$ as close as possible to the logits $(o_1, o_2, ..., o_n)$ from the frozen network $\mathcal{A}$. Following the method in [212], when m new classes are added at once, we compute the knowledge distillation loss by

$$L_{dist} = -\frac{1}{|\mathbf{X}^{c'}|} \sum_{x \in \mathbf{X}^{c'}}^{|\mathbf{X}^{c'}|} \sum_{k=1}^{n} \Big(p_k(x) * log[p'_k(x)] \Big) \tag{5.3}$$

where $p_k(x) = \frac{e^{o_k(x)/T}}{\sum_j^n e^{o_j(x)/T}}$ and $p'_k(x) = \frac{e^{o'_k(x)/T}}{\sum_j^n e^{o'_j(x)/T}}$, T is a temperature factor that is normally set to 2 [212]. $\mathbf{p} = \{p\}_n$ and $\mathbf{p}' = \{p'\}_n$ refer to the probabilities produced by the modified Softmax function in [214]. F_f and C_f denote the parameters of network $\mathcal{A}$. Similarly, F and C denote the parameters of network $\mathcal{B}$, as shown in Figure 5.1(b). $|\mathbf{X}^{c'}|$ indicates the number of images from the new m classes in a mini-batch. n denotes the number of the original classes. Note that n will be extended accordingly when more new classes are added.

5.3.3 Maximum mean discrepancy loss

Knowledge distillation loss focuses on constraining classification boundaries to mitigate the forgetting issue. However, for FGIR, it is more important to reduce the difference between feature distributions. For this, we adopt maximum mean discrepancy (MMD) [215] to capture the correlation of feature distributions between network $\mathcal{A}$ and $\mathcal{B}$. MMD has been used to bridge source and target distributions

such as in domain adaptation [222]. However, our work is the first to impose MMD to regularize the forgetting issue for FGIR.

Given the features R^d (d is feature dimension) from network $\mathcal{A}$ and $\mathcal{B}$, MMD measures the distance between the means of two feature distributions after mapping them into a reproducing kernel Hilbert space (RKHS). In Figure 5.2(a), we illustrate how MMD mitigates the catastrophic forgetting issue. Note that, in the Hilbert space $\mathcal{H}$, norm operation can be equal to the inner product [215]. Finally, the squared MMD distance is:

$$\begin{aligned}
&\mathrm{MMD}^2(\mathbf{R},\mathbf{R}') = ||\frac{1}{N}\sum_{i=1}^{N}\phi(R_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(R'_j)||^2_{\mathcal{H}} \\
&= \frac{1}{N^2}<\sum_{i=1}^{N}\phi(R_i) - \sum_{j=1}^{N}\phi(R'_j), \sum_{i=1}^{N}\phi(R_i) - \sum_{j=1}^{N}\phi(R'_j)>_{\mathcal{H}} \\
&= \frac{1}{N^2}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}<\phi(R_i),\phi(R_j)>_{\mathcal{H}} + \sum_{i=1}^{N}\sum_{j=1}^{N}<\phi(R'_i),\phi(R'_j)>_{\mathcal{H}} - 2\sum_{i=1}^{N}\sum_{j=1}^{N}<\phi(R_i),\phi(R'_j)>_{\mathcal{H}}\Big] \\
&\quad s.t.\ R = F_f(x),\ R' = F(x)
\end{aligned} \tag{5.4}$$

where N is batch size, and $\phi(\cdot)$ denotes the mapping function. However, it is hard to determine $\phi(\cdot)$. In RKHS, the kernel trick is used to replace the inner product in Eq. 5.4, *i.e.* $<\phi(R),\phi(R')>=k(R,R')$. Considering all the features in a mini-batch, $\mathbf{R}=\{R\}_N$ and $\mathbf{R}'=\{R'\}_N$, we define the MMD loss L_{mmd} as:

$$L_{mmd} = \mathrm{MMD}(\mathbf{R},\mathbf{R}') = \frac{1}{N}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}k(R_i,R_j) - 2\sum_{i=1}^{N}\sum_{j=1}^{N}k(R_i,R'_j) + \sum_{i=1}^{N}\sum_{j=1}^{N}k(R'_i,R'_j)\Big]^{\frac{1}{2}} \tag{5.5}$$

where $\mathrm{k(R,R')}=\exp(-(||\mathrm{R}-\mathrm{R}'||_2^2)/(2\sigma_{\mathrm{m}}^2))$, σ_m means m variances in the Gaussian kernel.

Discussion. Knowledge distillation loss focuses on constraining pair-wise similarity. However, MMD loss measures the distance between each feature vector, as depicted in Figure 5.2(b). Finally, it captures the distance of two feature distributions from the frozen net and the adaptive net. Thus, MMD loss is more powerful to quantize the correlation of two models.

Overall, the objective function in our method for incremental FGIR learning is:

$$L = \alpha L_{dist} + \beta L_{mmd} + (L_{ce} + L_{triplet}) \tag{5.6}$$

5.4 Experiments

5.4.1 Datasets and experimental settings

Datasets. We evaluate our method on the Stanford-Dogs [223] and Caltech-UCSD Birds-200 (CUB-200) [224] datasets. For the former, we use the official train/test splits. When training incrementally, we split the first 60 sub-categories (in the order of official classes) as the original classes and images from the remaining 60 sub-categories are added at once or sequentially. For the latter, we choose 60% of images from each sub-category as training set and 40% as testing set. Afterwards, we split the first 100 sub-categories (in the order of official classes) as the original classes and the remaining 100 sub-categories as new classes. The details are shown in Table 5.1.

Table 5.1: Statistics of the datasets used in our experiments.

Datasets	Training set (#Image/#Class)			Testing set (#Image/#Class)		
	Original cls.	New cls.	Total	Original cls.	New cls.	Total
Stanford-Dogs	6000/60	6000/60	12000/120	4651/60	3929/60	8580/120
CUB-200	3504/100	3544/100	7048/200	2360/100	2380/100	4740/200

Experimental settings. We use the Recall@K [131] (K is the number of retrieved samples), mean Average Precision (mAP), the precision-recall (PR) curve and feature distribution visualizations for evaluation. We adopt the Google Inception [62] to extract image features. During training, the parameters in Inception are updated using the Adam optimizer [225] with a learning rate of 1×10^{-6}, while parameters in fully-connected layers and classifier are updated with a learning rate of 1×10^{-5}. We follow the sampling strategy in [226] and each incremental process is trained 800 epochs. Following the practice in [131, 226], the output 512-D features (R^d) from fully-connected layers are used for retrieval. We set the hyper-parameters $\alpha = \beta = 1$ in Eq. 5.6, and the margin $\lambda = 0.5$ in Eq. 5.2.

5.4.2 One-step incremental learning for FGIR

We report the results for multiple classes added at once. The process includes two stages. First, we use the cross-entropy and triplet loss to train the network $\mathcal{A}$ on the original classes (100 classes for the CUB-200 dataset, 60 classes for the Stanford-Dogs dataset), denoted as $\mathcal{A}$(1-100) and $\mathcal{A}$(1-60), respectively. Second, only images of new classes are added at once to train network $\mathcal{B}$, denoted as $\mathcal{B}$(101-200) for CUB-200 and $\mathcal{B}$(61-120) for Stanford-Dogs. DIHN [218] has been explored the incremental learning for hashing-based image retrieval. However, its main difference with ours is to depend on the usage of old data as query set to avoid forgetting in their assumption. Considering no previous works for the fine-grained incremental image retrieval, we apply Learning without Forgetting (LwF) [212], Elastic Weight

Consolidation (EWC) [213], ALASSO [227], and the incremental learning for semantic segmentation (dubbed L2 loss) [219] for comparison. LwF, EWC, and ALASSO distill knowledge on classifier and network parameters, which are insufficient for incremental FGIR. L2 loss in [219] is more similar with ours where the knowledge is distilled on the classifier and intermediate feature space. Note that cross-entropy and triplet loss (*i.e.* semantic preserving loss) are combined with these three algorithms for fair comparison. The Recall@K results for the CUB-200 dataset are reported in Table 5.2.

Table 5.2: Recall@K (%) of incremental FGIR on the CUB-200 dataset when new classes are added at once. The best performance in the original class and the new class are in boldface.

Configurations	Original classes			New classes		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-100) (initial model)	79.41	85.64	89.63	-	-	-
+$\mathcal{B}$(101-200) w feature extraction	-	-	-	47.02	57.44	67.86
+$\mathcal{B}$(101-200) w fine-tuning	53.90	64.56	73.56	76.18	82.56	87.39
+$\mathcal{B}$(101-200) w LwF (*i.e.* L_{dist})	54.92	66.40	75.42	75.76	82.69	86.93
+$\mathcal{B}$(101-200) w ALASSO	56.91	66.65	76.57	72.48	79.50	85.67
+$\mathcal{B}$(101-200) w EWC	62.03	72.16	80.08	73.32	80.92	86.01
+$\mathcal{B}$(101-200) w L2 loss	66.48	75.68	82.67	**77.44**	**83.78**	**88.07**
+$\mathcal{B}$(101-200) w Our method	**74.41**	**82.57**	**88.52**	73.11	80.84	86.64
$\mathcal{A}$(1-200) (reference model)	77.33	85.08	89.03	76.64	83.53	89.12

In Table 5.2, the "w feature extraction" depicts when $\mathcal{A}$ directly extracts features on the new classes without re-training. The "w fine-tuning" depicts using L_{ce} and $L_{triplet}$ to train $\mathcal{A}$ on the new classes but without using L_{dist}. Overall, the network $\mathcal{B}$ suffers from the catastrophic forgetting issue and has lower performance on the original classes, whereas our method outperforms the others. As for the new classes, other three algorithms outperform ours. For example, " w L2 loss" method achieves on Recall@1 by 4.33% compared to ours (77.44%→73.11%). However, it suffers from significant performance degradation on the original classes with Recall@1 dropping by 12.93% compared to the initial model (79.41%→66.48%). For our method, the Recall@1 on the original classes is 74.41% (dropped by 5.00% from 79.41% of the initial model); the Recall@1 on the new classes is 73.11% compared to the reference model from $\mathcal{A}$(1-200) (*i.e.* Recall@1=76.64%). Similarly, the Recall@K results for the Stanford-Dogs dataset are reported in Table 5.3. We observe similar trends as the results we shown in main paper, when our method achieves good performance on the original classes and new classes with Recall@1= 76.67% and Recall@1=81.88%, respectively. Compared to the initial model on the original classes, our method has dropped Recall@1 performance by 4.00% (80.67%→76.67%).

We report the PR curves and mAP results in Figure 5.3(a), 5.3(b), and 5.3(c), respectively. Overall, when tested on the new classes, all methods share similar trends. When tested on the original classes, our method has better performance although it still has gap to reference performance. For mAP results, the reference

Table 5.3: Recall@K (%) of incremental FGIR on the Stanford-Dogs dataset when new classes are added at once. The best performance are in boldface.

Configurations	Original classes			New classes		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-60) (initial model)	80.67	87.27	92.20	-	-	-
+$\mathcal{B}$(61-120) w feature extraction	-	-	-	75.64	83.91	90.48
+$\mathcal{B}$(61-120) w fine-tuning	61.43	72.80	81.70	78.93	86.99	91.55
+$\mathcal{B}$(61-120) w LwF (*i.e.* L_{dist})	61.77	72.72	81.70	78.52	86.38	91.12
+$\mathcal{B}$(61-120) w EWC	62.24	73.30	82.82	78.90	86.59	91.19
+$\mathcal{B}$(61-120) w ALASSO	62.61	74.49	82.98	78.14	85.98	91.02
+$\mathcal{B}$(61-120) w L2 loss	72.07	81.44	87.47	**82.21**	88.75	92.52
+$\mathcal{B}$(61-120) w Our method	**76.67**	**85.10**	**91.14**	81.88	**88.98**	**93.36**
$\mathcal{A}$(1-120) (reference model)	79.29	86.86	91.61	82.57	88.75	93.13

results are the same as in Table 5.2. We utilize the well-trained network $\mathcal{A}$ at epoch=700 as initial model to train $\mathcal{B}$ on the new classes until convergence, we test the mAP of network $\mathcal{B}$ on the original classes. As the curves show, the network trends to degrade its accuracy on the original classes during incremental training. Similarly, we report the precision-recall curves and mAP results in Figure 5.4. We can observe these curves share with the similar trends with those from the CUB-200 dataset. Overall, our method can effectively address the catastrophic forgetting issue on the original classes while achieve ideal performance on the new classes.

Furthermore, we explore the influence of the number of the added new classes. Specifically, on the CUB-200 dataset, we choose 100 classes and 25 classes as new categories. The results are reported in Table 5.4. Likewise, for the Stanford-Dogs dataset, we choose 60 new classes and 5 classes for incremental learning, whose results are reported in Table 5.5. For CUB-Brids, we observe that larger newly-added classes lead to heavier forgetting. For example, when only 25 new classes are used, the Recall@1 drops from 79.41% to 76.65%, compared to the one drops from 79.41% to 74.41% where 100 new classes are added. Note that the reference models are trained jointly on all classes and tested on the original and new classes separately. For Stanford-Dogs, we observe these two datasets share with similar trends that larger new coming classes lead to heavier forgetting issue. For the Stanford-Dogs dataset, when only 5 new classes are added, the Recall@1 drops from 80.67% to 79.75%, compared to the one drops from 80.67% to 76.67% when 60 new classes are added.

We visualize the feature distributions using t-SNE [195] under two experimental settings: with and without MMD loss in Figure 5.5, which demonstrate the MMD loss reduces the distance between distributions and effectiveness for mitigating the forgetting issue.

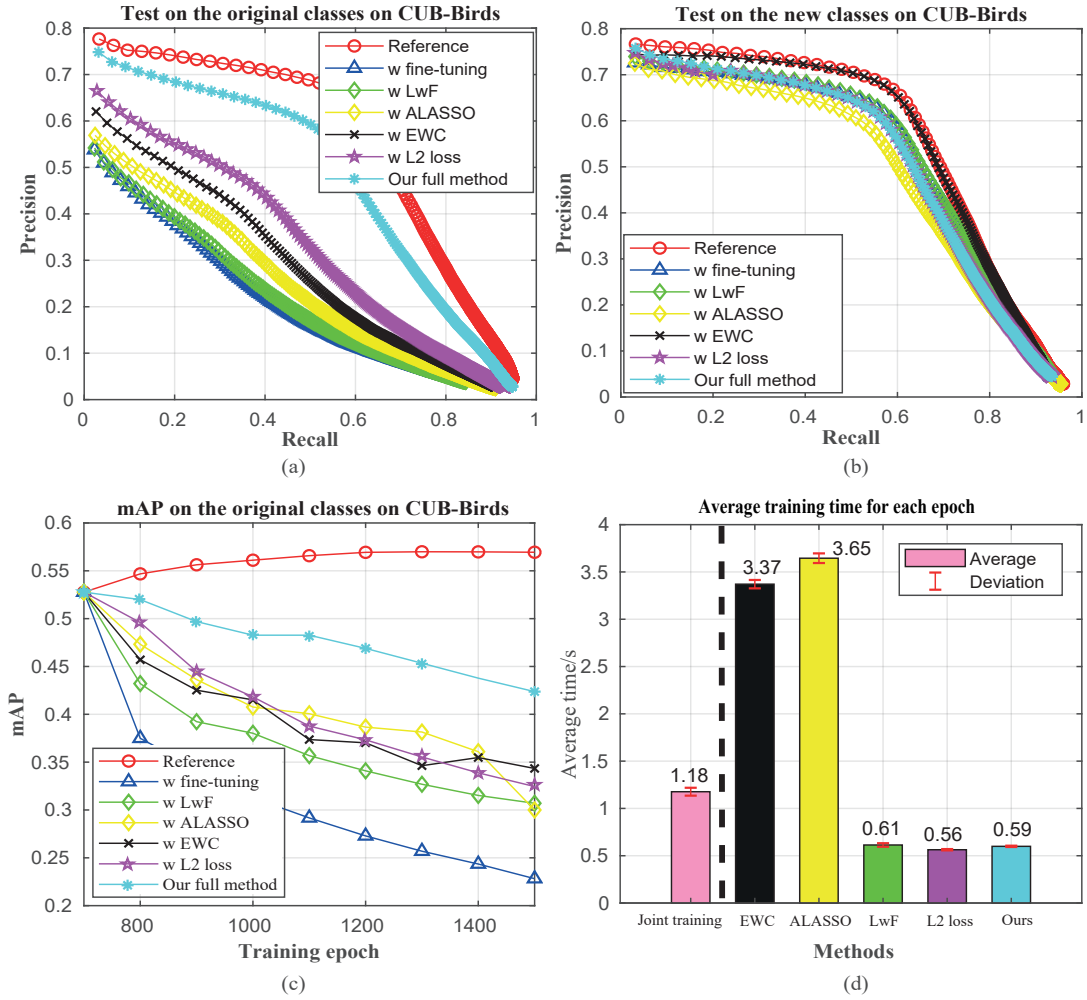


Figure 5.3: Performance evaluation on the CUB-200 dataset. (a)-(b) denote the PR curves tested on the original classes and new classes. (c) depicts the mAP results for different methods as the training proceeds. We only show the results tested on the original classes. (d) training time comparison during each epoch.

5.4.3 Multi-step incremental learning for FGIR

We split all new classes into 4 groups and added each sequentially. The training procedures are as follows: the initial model $\mathcal{A}$ is pre-trained on the original classes (1-100), and used as an initial model to train on newly-added classes (101-125) until

Table 5.4: Recall@K (%) on CUB-200 when 25 or 100 new classes are added at once. Correspondingly, † indicates the results are tested on different new classes.

Configurations	Original classes			New classes†		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-100) (initial model)	79.41	85.64	89.63	-	-	-
+$\mathcal{B}$(101-125) w Our method	76.65	83.47	88.86	73.13	82.31	88.44
+$\mathcal{B}$(101-200) w Our method	74.41	82.57	88.52	73.11	80.84	86.64
$\mathcal{A}$(1-125) (reference model)	77.84	83.94	87.80	79.25	85.54	91.96
$\mathcal{A}$(1-200) (reference model)	77.33	85.08	89.03	76.64	83.53	89.12

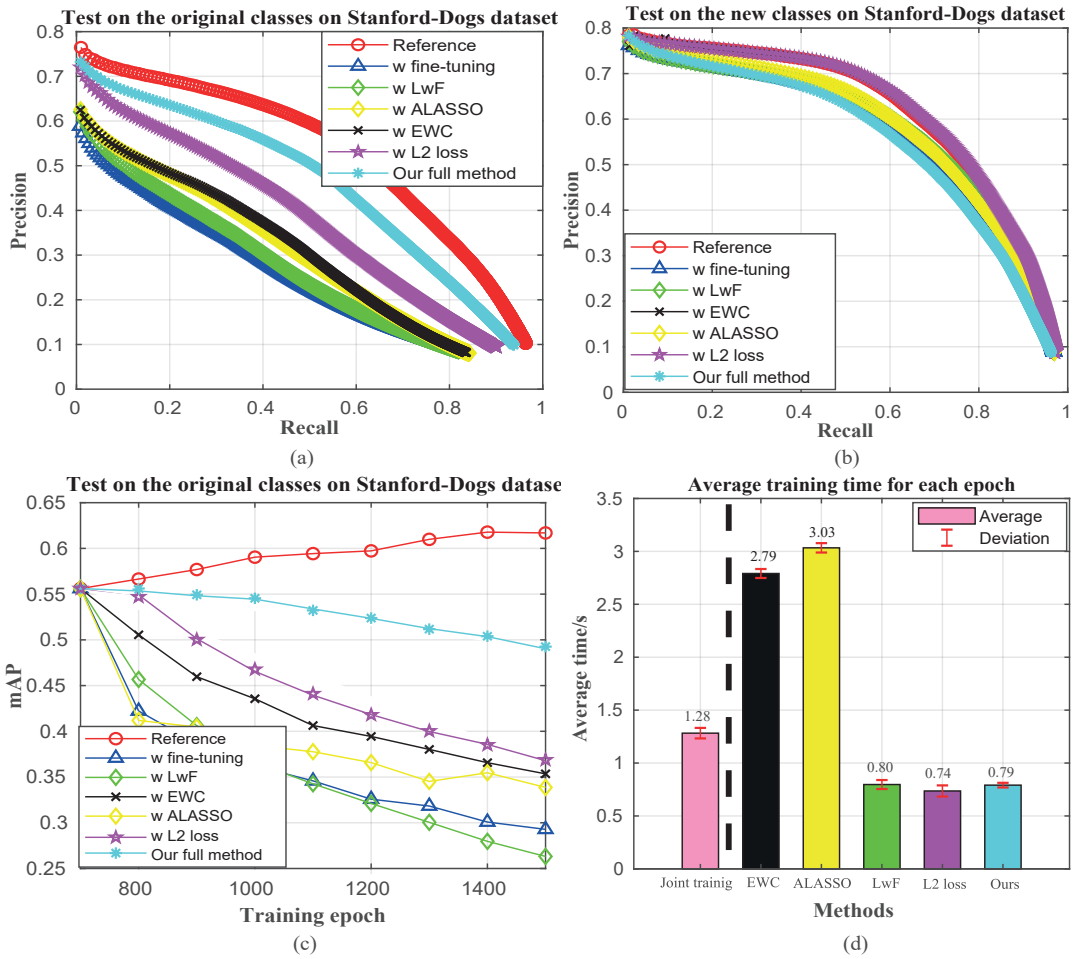
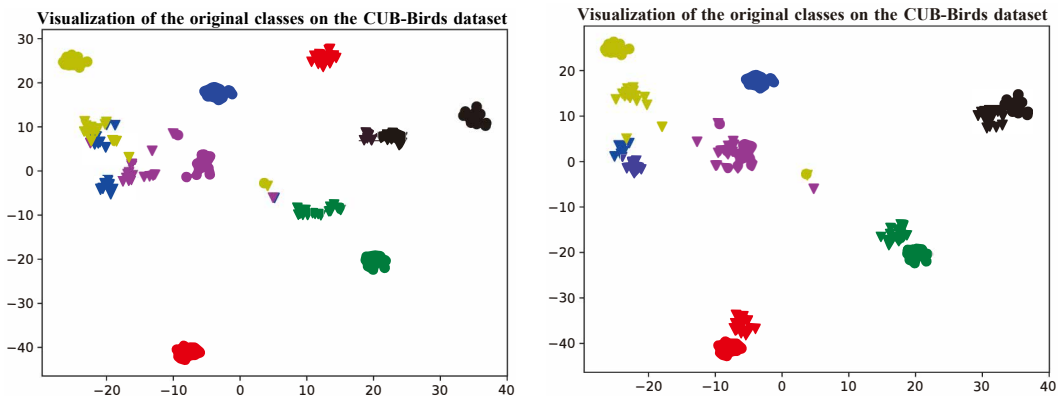


Figure 5.4: Performance evaluation on the Stanford-Dogs dataset. Figure (a)-(b) denote the precision-recall curves tested on the original classes and new classes. The larger the area under each curve, the better performance of the method. Figure (c) depicts the mAP results for different methods as the training proceeds. We only show the results tested on the original classes. Being closer to the reference curve (red one) indicates less performance degradation, *i.e.*, the method can maintain its previous performance on the original classes.

convergence to produce a new model $\mathcal{B}$(101-125). Afterwards, the newly-trained model $\mathcal{B}$(101-125) is used as an initial model to train on other new classes (126-150)

Table 5.5: Recall@K (%) on the Stanford-Dogs dataset when 5 or 60 new classes are added at once. Similar to the settings in Table 5.4, † indicates the results are tested on different new classes.

Configurations	Original classes			New classes†		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-60) (initial model)	80.67	87.27	92.20	-	-	-
+$\mathcal{B}$(61-65) w Our full method	79.75	87.23	91.92	97.45	98.55	99.27
+$\mathcal{B}$(61-120) w Our full method	76.67	85.10	91.14	81.88	88.98	93.36
$\mathcal{A}$(1-65) (reference model)	79.62	86.15	90.91	96.73	97.82	98.55
$\mathcal{A}$(1-120) (reference model)	79.29	86.86	91.61	82.57	88.75	93.13



(a) (b)

Figure 5.5: t-SNE visualization for feature distribution of 6 categories. The circle indicates the features from reference model, which has the same distribution in two cases. The triangle denotes the feature from models trained with/without MMD loss. (a): model trained without MMD loss; (b): model trained with MMD loss.

to produce $\mathcal{B}$(101-125)(126-150). This process is repeated until 4 groups of classes are added sequentially.

We compare to three representative methods (we choose EWC rather than ALASSO since EWC obtains higher performance on the CUB-200 dataset) and report the results in Table 5.9. The reference performances are achieved by jointly training all the classes, and then tested on each group (including the original classes). Overall, the model suffers from the catastrophic forgetting issue when sequentially training. However, our method achieves a minimal performance degradation. For instance, when 4 groups have been added, the model $\mathcal{B}$(101-125)(126-150)(151-175)(176-200) is tested on the original classes(1-100). The "L2 loss" algorithm Recall@1 drops 79.41%→67.37%→58.14%→53.86% → 45.85%, the average degradation is 8.39%. Our method Recall@1 drops 79.41%→76.65%→73.77%→70.47%→66.40%. The average performance degrades by 3.25%, which indicates that our method significantly mitigates the forgetting problem. Furthermore, our method has good performance on new classes, which are closer to the reference performance. When the model $\mathcal{B}$(101-125)(126-150)(151-175)(176-200) is tested on new classes (176-200), the results are achieved with Recall@1=85.21%, Recall@2=89.92% and Recall@4=93.28%, respectively, while the reference results are Recall@1=83.70%, Recall@2=90.25% and Recall@4=93.78%.

Similarly, we report the results on the Stanford-Dogs dataset in Table 5.10 when new classes are added sequentially. We observe similar trends as those for the CUB-200 dataset. Compared to the other two methods, the proposed method has ideal retrieval performance on the newly added classes and the original classes.

Table 5.6: Average top-1 accuracy of incremental learning for image classification on CIFAR-100 dataset [228].

Method	Number of the added new classes				
	20	40	60	80	100
L2 loss	77.3	47.5	40.5	36.6	32.8
EWC	77.3	60.5	50.9	43.3	39.5
LwF	77.3	62.5	52.9	46.2	41.0
Ours	77.3	64.6	55.8	49.2	43.3

Table 5.7: Ablation study for different components of loss function

Configurations	Original classes			New classes		
Recall@K(%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-100) (initial model)	79.41	85.64	89.63	-	-	-
+$\mathcal{B}$(101-200) w $L_{ce} + L_{triplet}$	53.90	64.56	73.56	76.18	82.56	87.39
+$\mathcal{B}$(101-200) w $L_{ce} + L_{triplet} + L_{dist}$	54.92	66.40	75.42	75.76	82.69	86.93
+$\mathcal{B}$(101-200) w $L_{ce} + L_{triplet} + L_{mmd}$	73.36	81.25	87.43	73.40	81.60	86.64
+$\mathcal{B}$(101-200) w $L_{ce}+L_{triplet}+L_{dist}+L_{mmd}$	74.41	82.57	88.52	73.11	80.84	86.64
$\mathcal{A}$(1-200) (reference model)	77.33	85.08	89.03	76.64	83.53	89.12

5.4.4 Validation with image classification

We evaluate the effectiveness of our method on CIFAR-100 [228] which is the popular benchmark for class-incremental learning in image classification. We split 100 classes into a sequence of 5 tasks, and each task includes 20 classes. In Table 5.6, the results indicate the average top-1 accuracy of the classes from seen tasks. In the last column, the test set evaluates the classes from all the five tasks. Note that, the 20 classes in the first task (the second column) achieve the same performance, as it has no incremental learning yet. We observe that our method outperforms other methods across the tasks. It suggests our method generalizes well to various applications. Notably, our improvement for image retrieval is more significant than that for image classification. The reason is that the proposed MMD loss is imposed on the feature representation, which largely benefits the metric learning for image retrieval. This also explains why our method is focused mainly on image retrieval.

Table 5.8: Sensitivity analysis of the hyper-parameters α, β. The better trade-off performance of the hyper-parameters are in bold face.

Configurations	Original classes			New classes		
Recall@K (%)	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-100) (initial model)	79.41	85.64	89.63	-	-	-
+$\mathcal{B}$(101-200) ($\alpha = 0.1, \beta = 0.1$)	56.53	66.31	75.59	77.52	83.82	88.15
+$\mathcal{B}$(101-200) ($\alpha = 0.1, \beta = 1$)	73.31	82.00	87.14	72.77	80.92	87.14
+$\mathcal{B}$(101-200) ($\alpha = 0.1, \beta = 10$)	79.58	85.76	90.47	49.50	61.51	70.59
+$\mathcal{B}$(101-200) ($\alpha = 1, \beta = 0.1$)	55.81	67.25	75.59	77.02	83.91	87.90
+$\mathcal{B}$(101-200) ($\alpha = 1, \beta = 1$)	**74.41**	**82.57**	**88.52**	**73.11**	**80.84**	**86.64**
+$\mathcal{B}$(101-200) ($\alpha = 1, \beta = 10$)	79.41	86.31	90.51	48.82	61.09	71.05
$\mathcal{A}$(1-200) (reference model)	77.33	85.08	89.03	76.64	83.53	89.12

5.4.5 Training time comparison

We compare the average training time on the CUB-200 dataset and the Stanford-Dogs dataset when 100 and 60 new classes are added at once. The results are shown in Figure 5.3(d) and 5.4(d), respectively. Note that all models in five methods are starting from the same initial model trained on the original 100 classes as initialization. The reference time is from joint training where the initial model is trained on all classes. The other four methods are incrementally learning the new classes only. We observe that our method saves more time by 50% as expected. EWC and ALASSO algorithms take more time than reference because the gradients computation during back-propagation process is time-consuming.

5.4.6 Components analysis

Ablation study. We have done an ablation study on the CUB-200 dataset when multiple classes are added at once. Note that the component "$L_{ce}+L_{triplet}$" comprises our baseline performance, thus we analyze the different loss items in Eq. 5.6. We can observe the influence of difference components for the original and new classes. The results are shown in Table 5.7.

Hyper-parameters sensitivity analysis. We explore the sensitivity of hyper-parameters α, β in Eq. 5.6, which affect significantly the trade-off performance. We conduct this experiment on the CUB-200 dataset. As shown in Table 5.8, we find that the incrementally-trained model is more sensitive to β than α. For instance, when α is set as 0.1, but β changes from 0.1 to 1, model $\mathcal{B}$ performs better on the new classes and significantly retains its previous performance. However, this obvious trend cannot be observed when β is set as 0.1, but α changes from 0.1 to 1 where the model $\mathcal{B}$ performs almost the same on the original and new classes. Finally, if $\alpha=\beta=1$, the incrementally-trained model $\mathcal{B}$ keeps a better trade-off performance between the original and the new classes.

5.5 Chapter Conclusions

In this chapter, for the first time, we have exploited incremental learning for fine-grained image retrieval in several scenarios for increasing numbers of image categories when only images of new classes are used. To overcome the catastrophic forgetting, we adopted the distillation loss function to constrain the classifier in the original network and the incremental classifier in the adaptive network. Moreover, we introduced a regularization function, based on Maximum Mean Discrepancy (MMD), to minimize the discrepancy between features of newly added classes from the original and the adaptive network. Comprehensive and empirical experiments on two fine-grained datasets show the effectiveness of our method that is superior over existing methods. In the future, it is promising to investigate incremental learning between different fine-grained datasets for image retrieval.

Table 5.9: Recall@K (%) results on the CUB-200 dataset when new classes are added sequentially. "Added new (101-125)" indicates the first 25 classes (101-125) are used as the first part to train the network.

Configurations		Original (1-100)			Added new (101-125)			Added new (126-150)			Added new (151-175)			Added new (176-200)		
Recall@K(%)		K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-100) (initial model)		79.41	85.64	89.63	-	-	-	-	-	-	-	-	-	-	-	-
with fine-tuning	+$\mathcal{B}$(101-125)	54.58	64.07	73.09	79.76	86.39	90.14	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)	41.53	53.18	63.55	61.56	73.64	84.01	70.33	80.00	88.00	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)	39.24	49.79	61.06	50.34	63.44	73.47	57.33	69.00	78.50	80.40	85.43	88.11	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)(176-200)	35.68	45.97	57.42	48.13	64.29	76.53	51.83	64.17	75.83	66.16	75.54	80.90	83.70	89.75	92.27
LwF algorithm [212]	+$\mathcal{B}$(101-125)	57.50	68.05	75.68	79.59	85.88	88.95	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)	42.46	54.03	64.66	62.59	74.83	82.31	70.17	79.67	86.00	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)	40.21	51.57	61.27	47.79	63.10	75.68	56.83	67.33	78.17	81.57	87.27	90.79	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)(176-200)	33.31	44.75	55.38	49.83	63.78	75.85	48.00	60.33	72.33	67.17	75.88	82.91	83.70	89.41	92.94
EWC algorithm [213]	+$\mathcal{B}$(101-125)	61.23	70.85	80.04	80.95	86.39	90.82	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)	46.65	56.40	67.54	65.48	77.72	84.01	72.33	80.67	86.67	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)	43.60	54.79	64.70	61.50	72.45	80.44	66.50	75.50	82.67	81.08	85.26	87.77	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)(176-200)	36.82	47.54	59.66	57.99	67.01	76.87	50.67	64.67	77.67	64.15	74.87	81.24	82.02	86.39	90.42
L2 loss algorithm [219]	+$\mathcal{B}$(101-125)	67.37	76.27	83.31	80.61	85.54	89.46	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)	58.14	68.31	76.78	72.11	80.44	87.41	73.33	82.17	88.67	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)	53.86	62.03	71.91	60.37	71.43	80.27	66.33	76.67	84.67	81.24	87.27	90.95	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)(176-200)	45.85	56.61	67.75	57.65	71.77	80.95	59.33	70.50	79.13	73.70	83.08	88.94	84.20	89.24	92.10
Our method	+$\mathcal{B}$(101-125)	76.65	83.47	88.86	73.13	82.31	88.44	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)	73.77	81.36	87.80	74.32	83.33	89.29	74.50	83.00	87.83	-	-	-	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)	70.47	78.77	85.97	70.41	80.78	88.78	72.00	79.17	86.83	78.89	86.77	90.26	-	-	-
	+$\mathcal{B}$(101-125)(126-150)(151-175)(176-200)	66.40	75.93	83.14	70.07	80.27	86.22	69.00	78.33	85.50	73.87	83.92	88.78	85.21	89.92	93.28
$\mathcal{A}$(1-200) (reference model)		77.33	85.08	89.03	76.87	84.86	90.48	73.00	80.00	87.67	83.25	88.94	92.29	83.70	90.25	93.78

Table 5.10: Recall@K (%) results on the Stanford-Dogs dataset when new classes are added sequentially. "Added new (61-75)" indicates we use first 15 classes (61-75) as the first incremental part to train the network.

Configurations		Original (1-60)			Added new (61-75)			Added new (76-90)			Added new (91-105)			Added new (106-120)		
Recall@K(%)		K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4	K=1	K=2	K=4
$\mathcal{A}$(1-60) (initial model)		80.67	87.27	92.20	-	-	-	-	-	-	-	-	-	-	-	-
with fine-tuning	+$\mathcal{B}$(61-75)	52.61	65.49	75.17	88.97	92.86	94.74	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)	40.81	53.32	64.78	69.92	79.95	86.72	79.74	88.17	93.21	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)	39.39	52.25	64.12	66.54	77.69	84.46	68.35	80.83	87.73	78.76	87.17	91.95	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)(106-120)	36.79	48.38	60.87	58.02	69.17	79.32	62.21	77.66	86.31	67.79	76.90	85.31	82.81	90.62	92.83
LwF algorithm [212]	+$\mathcal{B}$(61-75)	50.87	62.76	73.40	88.35	92.48	94.36	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)	42.06	53.62	65.10	71.18	82.46	88.10	77.33	87.19	92.44	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)	37.58	50.48	63.00	60.65	73.68	82.33	70.10	81.38	87.40	80.62	87.17	92.48	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)(106-120)	38.46	50.63	62.59	59.90	72.68	81.20	63.86	77.22	85.54	68.41	77.70	85.66	81.34	88.69	92.74
EWC algorithm [213]	+$\mathcal{B}$(61-75)	55.84	67.64	77.57	89.10	92.61	94.36	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)	45.32	58.29	68.85	79.82	85.21	90.35	81.38	88.72	93.32	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)	37.60	49.71	61.88	67.04	79.04	85.71	67.47	79.52	88.39	81.33	86.99	91.15	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)(106-120)	34.08	45.60	58.40	63.53	75.19	83.71	63.42	77.66	86.31	70.00	79.12	85.84	81.99	87.96	92.10
L2 loss algorithm [219]	+$\mathcal{B}$(61-75)	65.30	75.83	83.51	90.85	94.74	95.61	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)	55.97	67.36	77.04	84.46	90.73	92.73	80.94	89.38	93.54	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)	50.38	62.87	73.64	72.68	82.21	88.85	75.68	84.67	91.57	83.72	90.00	93.81	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)(106-120)	46.01	58.74	69.64	67.79	78.07	85.71	72.51	84.45	90.03	74.87	83.98	89.82	86.21	91.36	94.39
Our method	+$\mathcal{B}$(61-75)	76.07	84.88	90.11	91.85	95.36	96.87	-	-	-	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)	70.67	80.48	87.87	89.10	93.11	95.99	84.23	89.92	93.43	-	-	-	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)	67.75	79.17	86.45	86.09	91.98	95.49	81.60	90.25	93.76	84.25	89.03	93.45	-	-	-
	+$\mathcal{B}$(61-75)(76-90)(91-105)(106-120)	65.47	76.52	85.08	83.21	89.35	93.73	79.19	87.84	93.32	82.83	89.20	94.42	87.13	92.10	94.39
$\mathcal{A}$(1-120) (reference model)		79.29	86.86	91.61	92.61	94.99	96.37	82.48	90.80	93.76	83.72	91.33	95.58	86.12	93.11	95.96

Chapter 6

Feature Estimations based Correlation Distillation for Incremental Image Retrieval

In Chapter 5, we explored incremental learning for fine-grained image retrieval in which only the penultimate model is used for transferring previously learned knowledge. As incremental learning proceeds, each training session produces a specific model. Saving this stream of models will be memory-consuming. This raises a question that how to utilize the stream of models in incremental learning to transfer more previously learned information when learning on the current new data. We investigate this question by proposing a feature estimation method. Similar to the knowledge distillation framework in Chapter 5, we distill semantic correlations knowledge among the representations extracted from the new data only so as to regularize the parameters updates. In particular, for the case of learning multiple tasks sequentially, aside from the correlations distilled from the penultimate model, we estimate the representations for all prior models and further their semantic correlations by using the representations extracted from the new data. To this end, the estimated correlations are used as an additional regularization and further prevent catastrophic forgetting over all previous tasks, and it is unnecessary to save the stream of models trained on these tasks.

Keywords

Incremental learning, Fine-grained image retrieval, Correlations distillation, Feature estimation

This chapter is based on the following publication [38]:

- Chen, W., Liu, Y., Pu, N., Wang, W., Liu L., and Lew, M.S., "Feature Estimations based Correlation Distillation for Incremental Image Retrieval." IEEE Transactions on Multimedia, 2021.

6.1 Introduction

Learning is a life-long process for human beings so that we can learn continuously, devoid of forgetting previously acquired knowledge. However, this is not the case for deep neural networks, which suffer from the catastrophic forgetting problem [36]. Deep networks have been trained and validated for image retrieval on stationary datasets. As new data increase over time, the networks trained on the stationary datasets cannot be suited well for the non-stationary scenario.

The main challenge is to make the trained model adapt to new data without losing the knowledge on the seen data. Most conventional solutions for tackling this challenge suffer from obvious limitations. For example, joint training achieves optimal retrieval performance on old and new data, while it requires the presence of all the data. This is hard to meet for several scenarios where legacy data are unrecorded due to privacy issues or simply too cumbersome to collect old data. Moreover, retraining old data may lead to an imbalance issue between the quantity of old data and that of new data [229, 230].

Two incremental learning methods are developed to tackle the above limitations. First, the rehearsal based method utilizes generative adversarial nets to synthesize samples *w.r.t.* previous data distributions [231]. This method faces the difficulty of generating images with complex semantics. Second, the regularization based methods can either focus on network parameters or output activations. Parameters-based regularization methods estimate the parameter importance of previous tasks, then penalizes the drastic updates of these parameters when the model is learning a new task. Activation-based regularization methods, relying on the teacher-student framework, constrain the teacher model and the student model have similar outputs. The regularization methods have been explored for tasks such as image classification [229, 230, 232], but are less-explored for image retrieval. Recently, Parshotam *et al.* [233] regularize the representations via a normalized cross-entropy loss, training with metric learning for vehicle identification and retrieval. Chen *et al.* [37] propose regularizing both the representations and probabilities via the teacher-student framework for fine-grained image retrieval (FGIR) [234]. As depicted in Figure 6.1(a), they only use the penultimate model to transfer previously learned knowledge on old tasks.

For the case where new tasks are added sequentially, which is referred to multi-task incremental learning, only distilling on the penultimate model is insufficient to reduce forgetting on all previous tasks [235]. In fact, transferring additional knowledge learned on these tasks, *i.e.* via multi-model distillation tackles this insufficiency, as shown in Figure 6.1(b). In multi-task incremental learning, a stream of deep models is produced as new tasks are added continuously. However, it becomes too cumbersome and inefficient to store these models. Therefore, an arising question is that *how to use the model stream, not only the penultimate model, for knowledge distillation?*

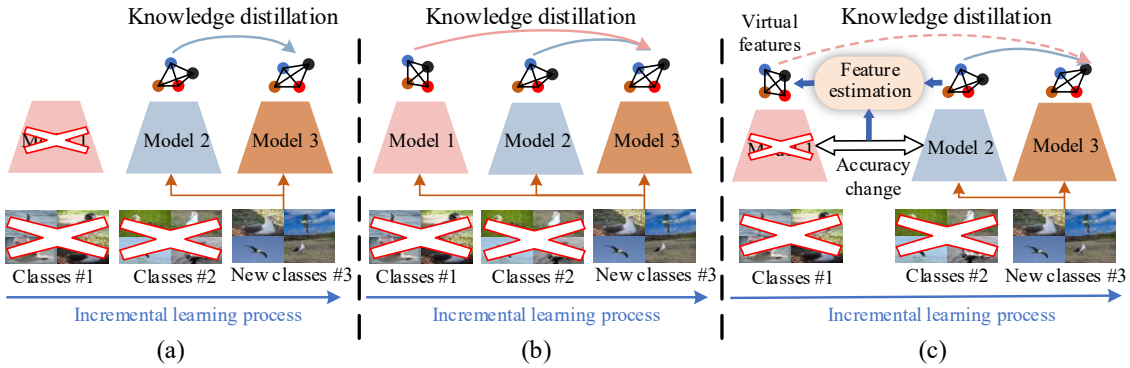


Figure 6.1: Comparison of three knowledge distillation methods. We depict three steps of distillation. (a) Single-model distillation method only stores and uses the penultimate model; (b) Multi-model distillation method has to store all old models and distills from them more knowledge devoid of forgetting; (c) Our method only stores the penultimate model while can accumulate previous knowledge learned at each model through feature estimations.

Few researchers address this problem in incremental tasks. Recently, a multi-model and multi-level knowledge distillation strategy is presented for incremental image classification [235]. However, the snapshots of all previous models still need to be saved and depend on network pruning methods to reconstruct.

In this chapter, we face the above question to improve deep model's continuous retrieval ability. Semantic correlations of features are transferred as knowledge from a teacher model to a student model when new data are used only. For multi-task incremental learning, the model stream trained on preceding tasks is unnecessarily saved. Instead, we estimate representations for these models and further their semantic correlations, using the features extracted from the current new task, as shown in Figure 6.1(c).

6.2 Related Work

Incremental image retrieval. Incremental learning can be categorized into architectural methods [230, 232], rehearsal methods [220, 231], and regularization methods [227, 235]. Most of them are used for image classification, regularizing the classification probabilities. Recently, incremental retrieval have been explored. CIHR [236] was proposed to deal with the concept drift issue for hashing retrieval in non-stationary environments. However, the selected images from previous training sessions are combined with new images to train hash tables. DIHN [218] is explored for incremental hashing retrieval where old data are used as a query set. Fine-grained incremental image retrieval is studied with only using new data [37]. However, knowledge is only transferred from the penultimate model, causing the insufficiency to remember previous knowledge when performing multi-task incremental learning. In this work, we further distill additional knowledge from the model stream

via a simple yet effective feature estimation method when only using new data in each incremental session.

Knowledge distillation. Knowledge can be distilled from the output of either the final classifier or the intermediate layers, relying on the teacher-student structures [237]. It is realized by characterizing the differences between the teacher model and the student model through metrics such as L1 distance [216], L2 distance [217], Gramian matrix [238], and KL-divergence [214]. For more details about knowledge distillation, we refer readers to a recent survey [237]. Knowledge distillation provides an effective way to retain the learned knowledge devoid of forgetting by one-teacher or a multi-teacher frameworks [235, 239]. For example, Zhou *et al.* [235] introduce using all previous models to transfer multi-level knowledge to train current new tasks. To avoid a great memory storage requirement, they prune previous models to get several "necessary" parameters during each training session.

Correlation learning has been used for multi-modal tasks to explore the relevance between different layers or data samples [240, 241, 242, 243, 244]. It focuses on the relations between feature representations rather than the features themselves. These relations enable models to explore rich contextual information of images such as [243] where three-level of correlations are integrated for optimal feature learning. Correlation learning can be combined into knowledge distillation. For example, Peng *et al.* [244] use a symmetric adjacency matrix to encode a knowledge graph with category correlations and transfer them via a semantic-visual mapping network. Similarity between activations of input pairs can also be extracted as knowledge to transfer into the student model [245]. The successful applications of correlation learning for knowledge distillation encourage its exploration for incremental learning tasks.

6.3 Correlations Distillation for Incremental Image Retrieval

6.3.1 Problem formulation

Given a dataset $\mathcal{D} = \{(\mathbf{X}^c, y^c)|c = 1, 2, \cdots, n\}$ with n classes, each of which c includes different amount of images $|\mathbf{X}^c|$ and they share the same ground-truth label y^c. The label is used to select a positive x_p and a negative x_n images for an anchor image x_a in each training iteration. A deep network $f(\cdot, \boldsymbol{\theta})$ learns representations $\boldsymbol{F} = f(\boldsymbol{X}, \boldsymbol{\theta})$ under the constraint of the triplet loss using hard sampling strategy, whose goal is to push away the distance $D(x_a, x_n) = ||f(x_a; \boldsymbol{\theta}) - f(x_n; \boldsymbol{\theta})||_2^2$ between x_n and x_a by a margin $\delta > 0$ compared to $D(x_a, x_p)$. Namely,

$$||f(x_a; \boldsymbol{\theta}) - f(x_p; \boldsymbol{\theta})||_2^2 + \delta < ||f(x_a; \boldsymbol{\theta}) - f(x_n; \boldsymbol{\theta})||_2^2 \tag{6.1}$$

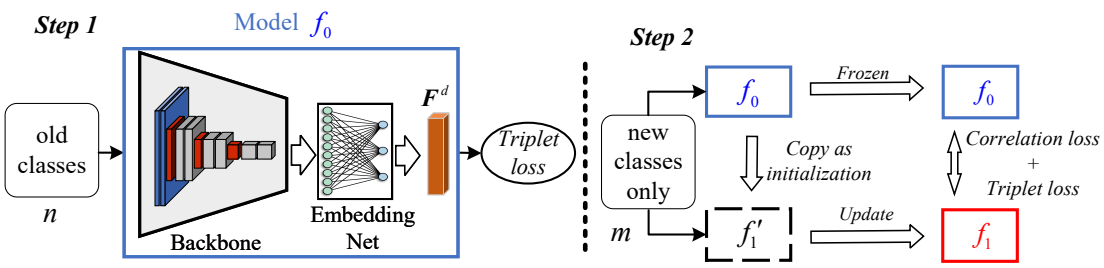


Figure 6.2: One-task incremental learning includes two training steps. **Step 1**: a model f_0 is well trained in advance on the n old classes using ranking loss only. **Step 2**: the well-trained model f_0 is frozen as a teacher network. Meanwhile, the parameters of the Backbone and the Embedding Net included in this model f_0 are copied as initialization for a temporary model $f_1^{'}$, which is updated to the final model f_1 under the constraints of correlation loss and triplet loss. At Step 2, only the m new classes are used for training.

Before incremental training, the network is well trained on the n old classes, converging at old parameters $\boldsymbol{\theta}_o$, *i.e.*,

$$\boldsymbol{\theta}_o = \underset{\boldsymbol{\theta}}{argmin}\ L_{triplet}(f_0(\mathbf{X}^c; \boldsymbol{\theta})) \tag{6.2}$$

where $L_{triplet}(x_a, x_p, x_n) = [\delta + D(x_a, x_p) - D(x_a, x_n)]_+$, as defined in Eq. 6.1. To train network f_0 incrementally, new data from m classes $\{(\mathbf{X}^{c'}, \mathbf{Y}^{c'})\}$ where $c' \in (n+1, n+2, ..., n+m)$ are added ($\{\mathbf{X}^c\} \cap \{\mathbf{X}^{c'}\} = \emptyset$) at once or sequentially, corresponding to one-task and multi-task cases, respectively.

The one-task case is depicted in Figure 6.2. At the start of training on m new classes, f_0 is copied into two copies. One is frozen as a teacher net, and another is used as a temporary initialization $f_1^{'}$ for further training ($\boldsymbol{\theta}_o = \boldsymbol{\theta}_n^{'}$, including the parameters in the Backbone and Embedding Net). We *only* use the m new classes to train to obtain f_1. Thus, the core issue of the one-task case is to make the model f_1 with new parameters $\boldsymbol{\theta}_n$ maintain a stable performance on the n old classes and achieve competitive accuracy on the m new classes. Formally, the overall objective for this scenario is:

$$L(\mathbf{X}^{c'}; \boldsymbol{\theta}_o; \boldsymbol{\theta}_n) = \underbrace{\lambda_1 L_{triplet}(\mathbf{X}^{c'}; \boldsymbol{\theta}_n)}_{\text{for plasticity}} + \underbrace{\lambda_2 L_{corr}(\mathbf{X}^{c'}; \boldsymbol{\theta}_o; \boldsymbol{\theta}_n)}_{\text{for stability}} \tag{6.3}$$

where $L_{triplet}$ makes the model perform well on new tasks while L_{corr} is the correlation loss to stabilize prior performance. $\boldsymbol{\theta}_o$ and $\boldsymbol{\theta}_n$ are the parameters for old tasks and new tasks, respectively. λ_1 and λ_2 are the plasticity and stability hyper-parameters, which tune the influence of two loss terms.

6.3.2 Correlations distillation for one-task scenario

As shown in Figure 6.2, the model $f_1^{'}$ serves as a to-be-trained student net. For the one-task incremental scenario, we propose to distill the semantic correlations as knowledge.

Specifically, the features with dimension d from the teacher model f_0 are formulated as $\boldsymbol{F}_o = f_0(\boldsymbol{X}^{c'}, \boldsymbol{\theta}_o) \in \mathbb{R}^{N\times d}$, and that from the student model f_1 are $\boldsymbol{F}_n = f_1(\boldsymbol{X}^{c'}, \boldsymbol{\theta}_n) \in \mathbb{R}^{N\times d}$. Based on the fact that semantically similar inputs produce similar patterns in a trained network [245]. Therefore, a Gram matrix with a kernel function $\mathcal{K}(\cdot)$ for $\boldsymbol{F}_o$ and $\boldsymbol{F}_n$ is defined:

$$G_o^{(i,j)} = \mathcal{K}(F_o^i, F_o^j);\ G_n^{(i,j)} = \mathcal{K}(F_n^i, F_n^j) \tag{6.4}$$

Here, we further define the function $\mathcal{K}(\cdot)$ as inner product, *i.e.*, $\mathcal{K}(F^i, F^j) = <F^i, F^j>$. Each entry (i, j) in $\boldsymbol{G} \in \mathbb{R}^{N\times N}$ represents the correlations of the same activation $(i = j)$ or these between different activations $(i \neq j)$. To compare the difference between $\boldsymbol{G}_o$ and $\boldsymbol{G}_n$, we first normalize these matrices with Softmax function $\sigma(\cdot)$, and then use KL-divergence to formulate a correlation loss L_{corr}.

$$L_{corr} = \frac{1}{N}\sum KL\big(\sigma(\boldsymbol{G}_o), \sigma(\boldsymbol{G}_n)\big) \tag{6.5}$$

6.3.3 Feature estimation for multi-task scenario

Compared to the one-task setting, the multi-task scenario is more complex where all m new classes are divided into t groups: $\boldsymbol{X}_0^{c'}$,..., $\boldsymbol{X}_t^{c'}$. For clarity, we illustrate its training process in Figure 6.3. As more new classes added sequentially, the model, correspondingly, evolutes from the initial model f_0 to the current one f_t. In practice, it may be difficult to save the stream of models. For this limit, we only save the model trained on the penultimate task $t-1$ when proceeding current task t for t^{th} new classes $\boldsymbol{X}_t^{c'}$. For example, when training on the 3rd group of new classes (task t=3), the knowledge is distilled only from the penultimate models f_2, while the previous models f_0 and f_1 are not saved. Due to the lack of previous models, it causes two drawbacks: (1) the knowledge is distilled only from the penultimate model f_{t-1} to the model on the current task t, and (2) the trained model f_t may forget more on old tasks prior to $t-1$. Therefore, it is natural to raise a question that how to utilize these unsaved models trained prior to the penultimate task $t-1$ for transferring additional knowledge to supervise the training of current task t.

Hereafter, for better understanding, we introduce the multi-task scenario by defining an adaptive model f_t for the current task t, a frozen model f_{t-1} trained on penultimate task $t-1$, and unsaved models $f_{t-2}, ..., f_0$ for earlier tasks $t-2, ..., 0$, as shown in Figure 6.4. Since the frozen model f_{t-1} is initialized from the previous unsaved model f_{t-2} at the start of training on task $t-1$, the feature distributions

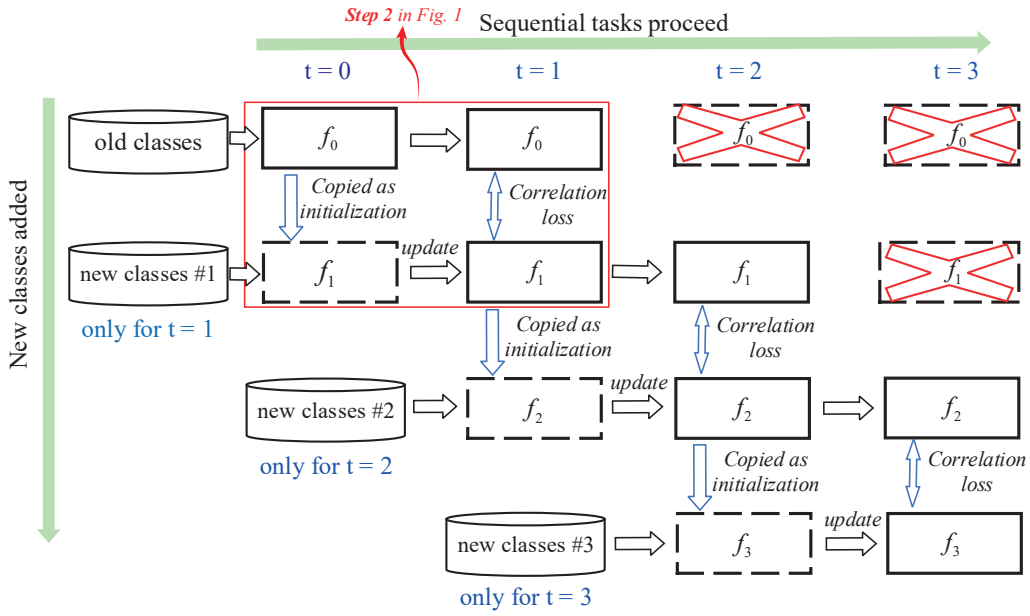


Figure 6.3: Illustration of multi-task incremental learning when three groups of new classes are added sequentially. For each round when new classes are added, the model trained on a previous task is frozen its parameters as a teacher net and is also copied as initializations of the model for new classes. Each round can be viewed as one-task incremental learning. For simplicity, the triplet loss is ignored.

of these two models have some inherent relations, which can be reflected through their accuracy (*e.g.*, mAP). This accuracy evolution along with training the models stream gives a hint for feature estimation.

a. Accuracy drops and accuracy gains

We propose a simple yet effective method to estimate the feature distributions for all unsaved models, which serve as an additional regularization term for training on current task t ($t \geq 2$). For this purpose, we first focus on the accuracy change during training from task $t-2$ to task $t-1$. Parameters of the penultimate model f_{t-1} are copied from those of the model f_{t-2}. Before training on task $t-1$, the accuracy on its old tasks and the new classes $\boldsymbol{X}_{t-1}^{c'}$ are recorded as Acc_o^b and Acc_n^b, respectively. Naturally, Acc_n^b is far from accurate since the penultimate model f_{t-1} is not trained specifically for new data. After training on task $t-1$, the accuracy on these old tasks and new classes $\boldsymbol{X}_{t-1}^{c'}$ are recorded as Acc_o^a and Acc_n^a, respectively. Intuitively, the model f_{t-1} acquires new knowledge on new classes $\boldsymbol{X}_{t-1}^{c'}$, and the accuracy increases from Acc_n^b to Acc_n^a (*i.e.*, accuracy gains). In contrast, model f_{t-1} may degrade accuracy from Acc_o^b to Acc_o^a (*i.e.*, accuracy drops) because this model is driven towards the new data.

The accuracy drops and accuracy gains, related to the stability-plasticity trade-off, are criteria that correspond to old tasks and new tasks, respectively. For instance, if a model has larger stability on previous tasks, both the accuracy drops and accuracy gains are small. In contrast, if the stability is too weak, the model suffers obvious

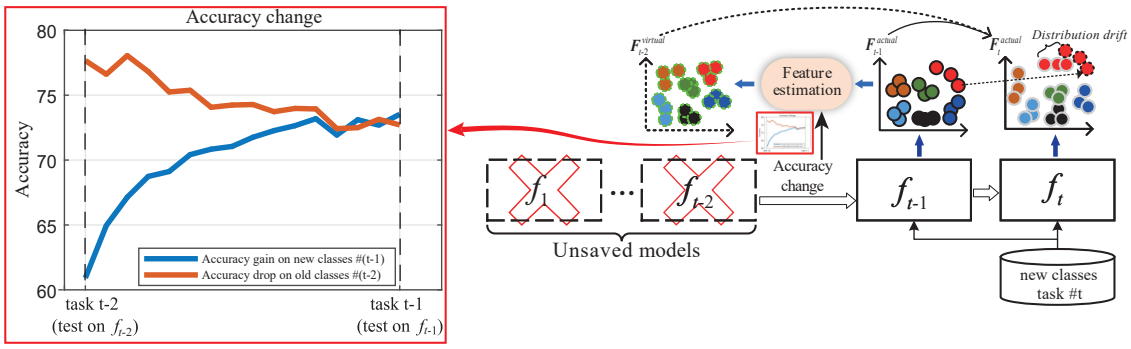


Figure 6.4: Illustration of feature estimation when performing the task t. The virtual feature distribution of unsaved model f_{t-2} can be estimated by that of frozen model f_{t-1} under multi-task incremental learning.

accuracy drops and forgetting on previous tasks. Inspired by [246], we define the accuracy changes using the accuracy drops and accuracy gains:

$$\alpha_{drop} = \frac{(Acc_o^a - Acc_o^b)}{Acc_o^b}, \alpha_{gain} = \frac{(Acc_n^a - Acc_n^b)}{Acc_n^b} \tag{6.6}$$

As the training proceeds from task $t-2$ to task $t-1$, their accuracy changes on old classes (the brown-color line in Figure 6.4) and new classes (the black-color line). Rather than saving these models, we only record their accuracy drops $\alpha_{drop}|_{(t-2)\rightarrow(t-1)}$ and accuracy gains $\alpha_{gain}|_{(t-2)\rightarrow(t-1)}$, which are meta-data of these models and provide implicit information to estimate the feature distribution drifts. Here, the subscript "$(t-2) \rightarrow (t-1)$" means the knowledge is distilled from task $t-2$ to penultimate task $t-1$.

b. Distribution drifts estimation

Estimating feature distribution drifts was explored in [247] where the attribute vectors are learned based on the source set and target set, then the learned vectors are used to estimate new features. In this work, we estimate feature drifts via the change of model accuracy. We only save the penultimate model f_{t-1} when training on current task t, see Figure 6.4. The recorded accuracy change from model f_{t-2} to model f_{t-1} has been reflected through the drifts of their feature distributions. Based on this, we use the accuracy change (α_{drop}, α_{gain}) and the available features from the model f_{t-1} to estimate the feature drifts which are used to further compute virtual features for model f_{t-2}. To be specific, when feeding t^{th} group of new classes $\boldsymbol{X}_t^{c'}$ into the model f_{t-1} and the adaptive model f_t, we obtain their corresponding actual features $\boldsymbol{F}_{t-1}^{actual} = f_{t-1}(\boldsymbol{X}_t^{c'})$ and $\boldsymbol{F}_t^{actual} = f_t(\boldsymbol{X}_t^{c'})$. Since the accuracy drops and accuracy gains from model f_{t-2} to model f_{t-1} have been obtained, we estimate their feature distribution drifts using a simple yet effective method:

$$\begin{aligned}
&\boldsymbol{\Delta}|_{(t-2)\to(t-1)} \approx \boldsymbol{\alpha} \cdot \boldsymbol{F}_{t-1}^{actual} \\
&s.t.\ \boldsymbol{\alpha} = Cat(\alpha_1, ..., \alpha_i, ..., \alpha_N),\ \alpha_i \in \mathbb{R}^d, \boldsymbol{\alpha} \in \mathbb{R}^{N\times d} \\
&\alpha_i \sim U(\alpha_{drop}|_{(t-2)\to(t-1)}, \alpha_{gain}|_{(t-2)\to(t-1)})
\end{aligned} \tag{6.7}$$

where $Cat(\cdot)$ means vector concatenation operation. Each raw vector α_i is randomly sampled from the uniform distribution $U(\cdot,\cdot)$ according to α_{drop} and α_{gain}. Thereby, $\boldsymbol{\alpha}$ has the same dimension with the features $\boldsymbol{F}$. In theory, the expectation of each sampling in $\boldsymbol{\alpha}$ is close to $0.5 \times (\alpha_{drop} + \alpha_{gain})$.

It is assumed that the features change uniformly during sequential training and the changes can be reflected through the accuracy drops and accuracy gains. With this hypothesis, the feature drifts $\boldsymbol{\Delta}|_{(t-2)\to(t-1)}$ can be evaluated according to the actual features $\boldsymbol{F}_{t-1}^{actual}$. With the feature drifts, inspired by [247], the virtual feature distributions for unsaved model f_{t-2} are estimated:

$$\boldsymbol{F}_{t-2}^{virtual} = \boldsymbol{F}_{t-1}^{actual} + k\,\boldsymbol{\Delta}|_{(t-2)\to(t-1)} \tag{6.8}$$

where k is a scaling factor, we set $k = 1$. The reason why we can estimate the virtual features $\boldsymbol{F}_{t-2}^{virtual}$ from $\boldsymbol{F}_{t-1}^{actual}$ is because the parameters of model f_{t-1} are initialized from model f_{t-2} at the start of training f_{t-1}.

Similarly, we can further approximate the virtual feature $\boldsymbol{F}_{t-3}^{virtual}$ for model f_{t-3} according to the already-estimated $\boldsymbol{F}_{t-2}^{virtual}$, its accuracy drops $\alpha_{drop}|_{(t-3)\to(t-2)}$ and accuracy gains $\alpha_{gain}|_{(t-3)\to(t-2)}$ from task $t-3$ to task $t-2$. Normally, with a recursive scheme, the virtual features of all previous unsaved models can be estimated using their recorded accuracy drops, accuracy gains, and already-estimated virtual features. Finally, the features of first model f_0 are estimated as:

$$\begin{aligned}
\boldsymbol{F}_0^{virtual} = &(1 + k\,\boldsymbol{\alpha}|_{(t-2)\to(t-1)})(1 + k\,\boldsymbol{\alpha}|_{(t-3)\to(t-2)}) \\
&...(1 + k\,\boldsymbol{\alpha}|_{(0)\to(1)})\boldsymbol{F}_{t-1}^{actual}
\end{aligned} \tag{6.9}$$

c. Importance for estimated features

The estimated features for all previous unsaved models serve as additional regularization terms. Thus, more Gram matrices $\boldsymbol{G}^{virtual}$ are computed based on these estimated features, as illustrated in Figure 6.5. To this end, the additional correlation loss, such as $L_{corr}^{(t-2)\to t}$, based on the estimated features is formulated as:

$$L_{corr}^{(t-2)\to t} = \frac{1}{N}\sum\Big(KL\big(\sigma(\boldsymbol{G}_{t-2}^{virtual}), \sigma(\boldsymbol{G}_{t}^{actual})\big)\Big) \tag{6.10}$$

When more new classes are added sequentially, more Gram matrices are computed through the recursively-estimated features. However, these Gram matrices cannot be treated identically when used for regularizing the current task t since the accumulated errors may make the recursively estimated features more and more unreliable. For this limitation, the estimations for earlier tasks are assigned with a smaller importance. Naturally, the importance is related to the indices of old tasks. Finally, we formulate the correlation loss terms with different importance factors:

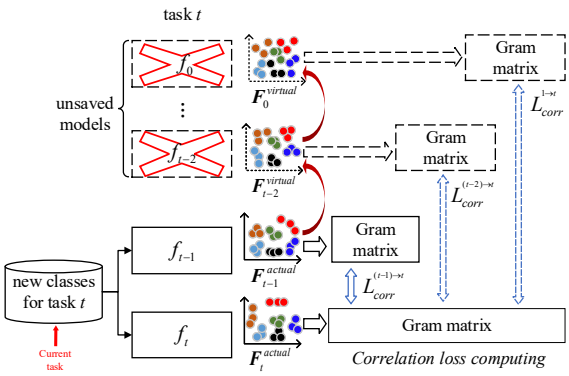


Figure 6.5: Feature estimations based knowledge distillation. The red arrows denote feature estimation process. The dash arrows indicate the features are virtually estimated from the actual features. For instance, the superscript "$(t-2) \rightarrow t$" refers to the Gram matrix of task $(t-2)$ is used as supervision for training current task t, which is formally defined in Eq. 6.5.

$$L_{corr} = L_{corr}^{(t-1)\rightarrow t} + \underbrace{\frac{1}{(t-1)} L_{corr}^{(t-2)\rightarrow t} + \frac{0.1}{(t-2)} L_{corr}^{(t-3)\rightarrow t} + \ldots + \frac{(0.1)^{t-2}}{1} L_{corr}^{1\rightarrow t}}_{\text{Feature estimation for prior sequential tasks(t}\geq 2)} \tag{6.11}$$

For one-task incremental scenario (t=1), Eq. 6.11 can be re-written as Eq. 6.5. If more tasks are performed ($t \geq 2$), each semantic correlation loss based on the estimated virtual features are constrained with importance factors ($\frac{1}{(t-1)}$, $\frac{0.1}{(t-2)}$,...). Substituting the term Eq. 6.11 into Eq. 6.3, we obtain the overall objective function for incremental FGIR.

6.4 Experiments

6.4.1 Datasets and experimental setup

We evaluate the method on two datasets: Caltech-UCSD Birds-200 (CUB-200) [224] and Stanford-Dogs-120 (Dogs-120) [223]. We choose 60% images from each category as training sets and 40% as testing sets. Afterwards, we split the first 100 categories (60 for the Dogs dataset) as the old classes (*i.e.*, n=100 or 60) and the remaining 100 (60 for the Dogs dataset) categories as new classes (*i.e.*, m=100 or 60). For the multi-task case, these new classes are divided into several groups evenly. All splits are in the order of official classes. In the following text, we use the class index of each dataset to denote a group of new classes. For example, "*classes (101-125)*" in italic means that the m=25 new classes from the index 101 to 125 are used for training, the corresponding trained model is f_1*(101-125)*. The details of datasets are followed the splitting methodology in Table 5.1 in Chapter 5.

Implementation details. We utilize Google Inception as a backbone net. The whole process includes two stages: initial training and incremental training. In the first stage, the initial model f_0 is trained on the n old classes by using the Adam optimizer with a learning rate of 1×10^{-6}, its embedding net is updated with a learning rate of 1×10^{-5}. In the second stage, we train a new model f_1 based on the converged f_0 on the m new classes using Eq. 6.3, with the same learning rate in the first stage. The model f_0 trained on the n old *classes (1-100)* or *(1-60)* is wrote as f_0*(1-100)* or f_0*(1-60)*. Likewise, the model f_1 is represented by the added m new classes, such as f_1*(101-200)* or f_1*(61-120)*. Following the practice in [131, 226], the output 512-d features (F^d in Figure 6.2) are used for retrieval[1]. We set the plasticity factor $\lambda_1 = 1$ and stability factor $\lambda_2 = 10$ in Eq. 6.3 for the following experiments.

Evaluation metrics. We use the Recall@1 [131, 248] and mean Average Precision (mAP) as retrieval metrics, and use average incremental accuracy [232, 249] and average forgetting [246] to evaluate incremental learning.

Table 6.1: Recall@1 and mAP (%) of incremental FGIR trained for the one-task scenario, "Initial model f_0*(1-100)*" indicates model trained on the first 100 classes on the CUB-200 datasets. "Reference model" indicates the model f_0*(1-200)* trained on all classes of CUB-200. The best performance is reported in boldface.

Dataset	**Caltech-UCSD Birds-200**					
Configuration and Results (%)	Old *classes (1-100)*		New *classes (101-200)*		Average	
	Recall@1	mAP	Recall@1	mAP	Recall@1	mAP
Initial model f_0*(1-100)*	79.24	55.78	46.93	19.54	63.09	37.66
⇒ Model f_1 w fine-tuning	70.21	42.57	**75.13**	**48.90**	72.67	45.74
⇒ Model f_1 w EWC [213]	73.32	45.73	72.84	44.14	73.08	44.94
⇒ Model f_1 w ALASSO [227]	72.88	43.87	72.94	45.50	72.91	44.69
⇒ Model f_1 w NCE_{EWC} [233]	72.63	43.80	73.07	45.15	72.85	44.48
⇒ Model f_1 w L2 loss [219]	75.93	50.23	74.12	47.47	75.03	48.85
⇒ Model f_1 w MMD loss [37]	77.03	51.10	74.12	45.05	75.58	48.08
⇒ Model f_1 w Our method	**77.71**	**52.25**	75.00	46.51	**76.36**	**49.38**
Reference model f_0*(1-200)*	78.18	52.17	79.24	50.99	78.71	51.58

6.4.2 One-task scenario evaluation

Baselines. CIHR [236] and DIHN [218] have been explored for incremental hashing retrieval. The main difference with ours is that they used old data for training, while we use new data *only*. For a fair comparison, we take [37] as a baseline in which the feature-level regularization (*i.e.*, maximum mean discrepancy (MMD) loss) is used. We also compare to the popular algorithms including EWC[2], ALASSO[3], NCE loss[4], and L2 loss. Specifically, EWC [213] and ALASSO [227] are the network parameters

[1]Code available at: https://github.com/cw1091293482/Deep-Incremental-Image-Retrieval
[2]https://github.com/joansj/hat/tree/master/src/approaches
[3]https://github.com/dmpark04/alasso
[4]https://github.com/ProsusAI/continual-object-instances

Table 6.2: Recall@1 and mAP (%) of incremental FGIR trained for the one-task scenario, "Initial model f_0*(1-60)*" indicates model trained on the first 60 classes on the Stanford-Dogs datasets. "Reference model" indicates the model f_0*(1-120)* trained on all classes of Stanford-Dogs. The best performance is reported in boldface.

Dataset	**Stanford-Dogs-120**					
Configuration and Results (%)	Old *classes (1-60)*		New *classes (61-120)*		Average	
	Recall@1	mAP	Recall@1	mAP	Recall@1	mAP
Initial model f_0*(1-60)*	81.27	66.05	69.28	34.13	75.28	50.09
⇒ Model f_1 w fine-tuning	73.96	45.24	**83.69**	**67.25**	78.83	56.25
⇒ Model f_1 w EWC [213]	74.76	46.92	81.45	62.69	78.11	54.81
⇒ Model f_1 w ALASSO [227]	75.92	48.35	81.50	63.40	78.71	55.88
⇒ Model f_1 w NCE$_{EWC}$ [233]	75.12	47.88	81.62	62.99	78.37	55.44
⇒ Model f_1 w L2 loss [219]	78.99	56.57	83.23	66.63	81.11	61.60
⇒ Model f_1 w MMD loss [37]	79.49	**59.43**	83.35	65.21	81.42	**62.32**
⇒ Model f_1 w Our method	**79.92**	58.37	83.48	66.01	**81.70**	62.19
Reference model f_0*(1-120)*	80.37	62.48	83.10	66.78	81.74	64.63

regularization methods. To deploy these methods, we further train a classifier on the top of the embedding net. NCE loss [233] regularizes the inner product of an anchor-positive pair and anchor-negative pairs via a normalized cross-entropy loss. This method is combined into EWC algorithm. We follow this protocol by mining 9 hard negative samples (termed as NCE$_{EWC}$). L2 loss [217] focuses on minimizing the Euclidean distance between the features from the teacher-student models. For a fair comparison, the above four methods are trained with triplet loss $L_{triplet}$, having the same hyper-parameter $\lambda_1 = 1$. In terms of the plasticity factor λ_2, we tune this factor for four methods in incremental FGIR until we get their optimal performance. As a result, the corresponding plasticity factors are tuned as 8000, 0.2, 10, and 0.1, respectively. Moreover, the "Reference" by joint learning serves as an *upper-bound* performance. The fine-tuning method is also used as a reference for the new tasks since there is no knowledge distillation regularization.

One-task incremental learning (m=100 or m=60) is similar to transfer learning, while incremental training further emphasizes reducing forgetting on the n old classes. The results are reported in Tables 6.1 and 6.2. Note that only model f_0 is available, thereby it is unnecessary to estimate virtual features.

Naturally, the initial model f_0 trained on the n old classes performs poorly on the m new unseen classes. Take the CUB-200 dataset as an example, mAP is 19.54% when the initial model f_0*(1-100)* is tested on the m new classes without any re-training. Using the initial model f_0, we further re-train on the m new classes using different incremental algorithms to obtain the model f_1, whose performance is distinct on the old and new classes, as shown in Table 6.1. The fine-tuning method achieves the best accuracy on the new classes, it improves the accuracy (19.54%→48.90% in mAP) on the new classes on the CUB-200 dataset but degrades accuracy (*i.e.*, forgetting)

on the old classes (55.78%→42.57% in mAP). Similar trends can be observed on the Stanford-Dogs dataset in Table 6.2.

For other algorithms, the models trained by network parameters regularization methods such as EWC and ALASSO show a similar trend that they reduce forgetting on the n old classes, but their performance on the m new classes is less competitive compared to the fine-tuning method. NCE_{EWC} regularises metric learning via cross-entropy loss on the feature embeddings. We find this method has some limited benefits. For example, it improves on the Stanford-Dogs dataset in terms of the average performance. L2 loss and MMD loss regularize the features directly. For L2 loss, it regularizes the model f_1 to forget less on the old classes of two datasets. For instance, on the CUB-200 dataset, it reduces the degradation by 3.31% of Recall@1 (79.24%→75.93%) and 5.55% of mAP (55.78%→50.23%), see Table 6.2 and Table 6.1 for details.

MMD loss is more similar to our method in which feature correlations are also considered [37]. Compared to MMD loss, our method, in most cases, suffers less accuracy degradation on two datasets. For instance, our method degrades the Recall@1 on the n old classes by 1.53% (79.24%→77.71%) and 1.35% (81.27%→79.92%) on CUB-200 and Stanford-Dogs, respectively, whereas the MMD loss degrades the Recall@1 on the old classes by 2.21% (79.24%→77.03%) and 1.78% (81.27%→79.49%) on two datasets. Moreover, in terms of the performance on the m new classes, our method also achieves closer accuracy to that of the fine-tuning method.

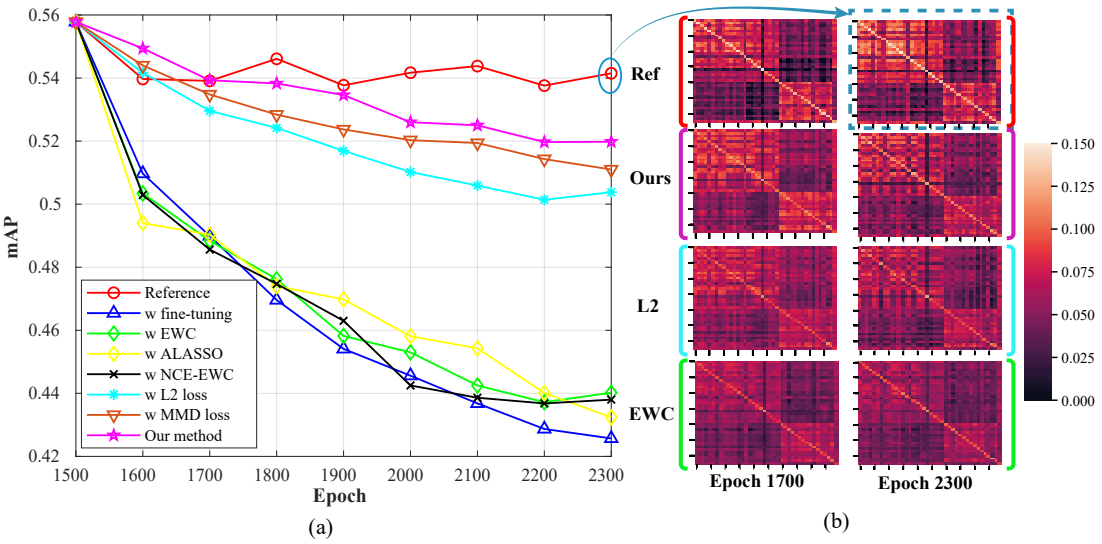


Figure 6.6: (a) mAP evolution of old classes *(1-100)* tested on the CUB-200 dataset under one-task scenario. (b) The Gram matrices of four representative methods (best viewed in color). More brightness indicates higher semantic correlations between two samples. The reference performance is obtained by joint training. Our method retains most semantics (higher brightness) compared to EWC and L2 loss.

Besides, we report the mAP evolution during incremental training in Figure 6.6(a). The activation regularization methods (*e.g.*, L2 loss) outperform the network param-

eters regularization methods (*e.g.*, EWC). Moreover, we visualize the Gram matrices of three methods. As the training proceeds, their differences with respect to the reference Gram matrices are maximized. Namely, the bright area in the three methods becomes ambiguous. However, our method retains most semantics of old classes (more brightness) than the other two continual learning strategies even at the last training epoch.

6.4.3 Multi-task scenario evaluation

Multi-task scenario refers to the case that m new classes are divided evenly into several groups and added sequentially. For the CUB-200 dataset, the remaining 100 new classes are split into 4 disjoint groups, with 25 classes per group; For the Stanford-Dogs dataset, we also get 4 groups with 15 classes per group. Thus, there are 4 steps incremental training for each dataset. For each step, the model is trained only on the images from a new class group (*e.g.*, *classes (126-150)* of the CUB dataset) and is tested separately in prior groups (*e.g.*, *classes (1-100)* and *classes (101-125)*) to evaluate the forgetting rate of this step. Note that incremental performance is *insensitive* to the arrival order and choice of new classes since the tasks do not depend on softmax-based probabilities [250].

Accuracy change range. We estimate the features of previous models (using Eqs. 6.7 and 6.8) based on the accuracy change defined in Eq. 6.6. Concretely, we use mAP to calculate the accuracy range. For instance, on the CUB-200 dataset, model f_0*(1-100)* takes as input the first group of new classes (see Figure 6.3) and produces an incrementally-trained model f_1*(101-125)*. In terms of mAP, it degrades from 54.20% to 52.44% on the $n = 100$ old classes while increases from 29.82% to 52.27% on the $m = 25$ new classes. These recorded mAPs are used to calculate the accuracy change range $(\alpha_{drop}, \alpha_{gain})$ using Eq. 6.6. Finally, the mAP change range is (-0.0325, 0.7528) during task $t = 1$ and is used to estimate the features for model f_0 when training the next task $t = 2$, without storing this model. The estimated features serve as an extra regularization for training task $t = 2$ in which the knowledge is mainly transferred from the model f_1*(101-125)* to f_2*(126-150)*. This process is performed repeatedly until all new class groups are added. The earlier feature estimation procedure becomes less reliable as more groups of new classes are added. We solve this issue by decreasing importance factors in Eq. 6.11. For multi-task scenario, we keep the plasticity factor λ_1 and stability factor λ_2 in Eq. 6.3 as 1 and 10, respectively.

We adopt forgetting measurement [246] to quantify the forgetting ratio. Specifically, the forgetting ratio for a particular task is defined as the difference between the maximum accuracy gained throughout the incremental training process in the past and the accuracy the currently-trained model has, then all t tasks forgetting ratios are averaged:

$$forgetting = \frac{1}{t-1}\sum_{j=1}^{t-1}\Big(\max_{l\in\{1,\dots,t-1\}} Acc_{l,j} - Acc_{t,j}\Big), \forall j < t \tag{6.12}$$

where $Acc_{t,j}$ denotes the accuracy of jth group of new classes evaluated by the model trained on the task t. Concretely, we employ the mAP metric as Acc for evaluation. When the model has been incrementally trained up to task t, we measure and then average all previous forgetting ratios $(1, 2, ..., t-1)$ using Eq. 6.12 as final forgetting evaluation.

The average forgetting ratios are depicted in Figure 6.7. Note that we use the task index to indicate the group of new classes being added. For example, "$t = 2$" on the CUB-200 dataset means the model is training on the 2nd group of new classes and then tested on *classes (1-100)* and *classes (101-125)* separately. Obviously, all methods suffer catastrophic forgetting on two datasets. In particular, fine-tuning on a new task leads to significant forgetting on the old tasks. EWC and ALASSO cannot reduce the forgetting issue ideally in the multi-task scenario. By contrast, activation regularization methods perform better on two datasets. Particularly, MMD loss and our method, by distilling feature correlations, can significantly reduce the forgetting ratio compared to the L2-regularized feature alignment method. Our method can further largely mitigate the forgetting ratio when feature estimation is considered into correlations distillation. Finally, our method has the least forgetting ratio (up to 10%) on these two datasets.

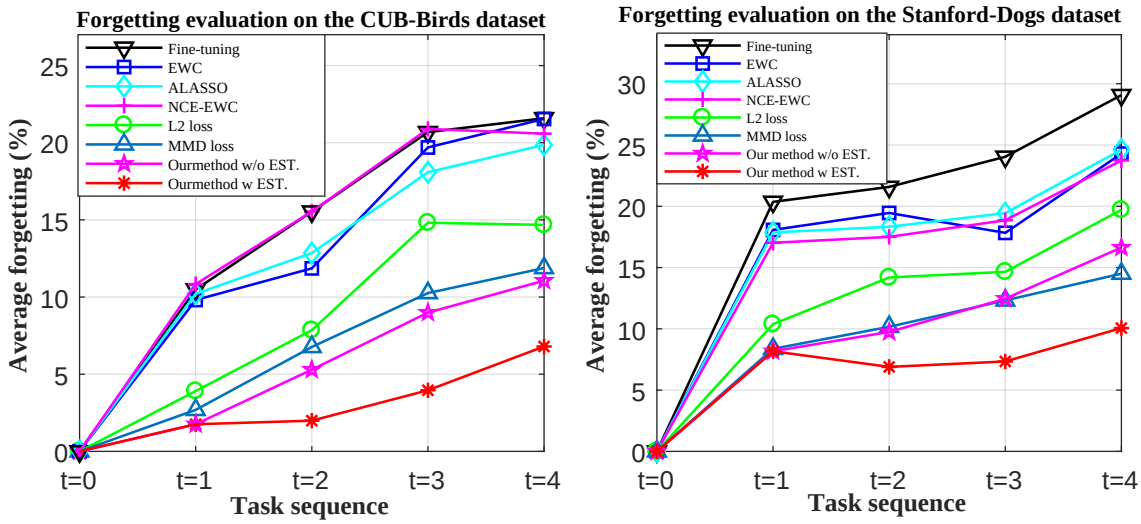


Figure 6.7: Average forgetting evaluation. "w/o EST." indicates that feature ESTimation strategy is not included in our method (see Eq. 6.11). The forgetting is measured on previous old classes after training on current new classes. The forgetting ratios over all previous tasks are averaged to show. The higher value indicates the more severe forgetting.

After all new tasks are added sequentially (*i.e.* $t = 4$), we get the final model

f_4*(176-200)* or f_4*(106-120)* for this task. We measure the accuracy of each prior task (*i.e.*, class group) using the final model. We take Recall@1 as a metric for demonstration, as shown in Figure 6.8, including the performance for the previous tasks and the last new task. In this experiment, we use the performance of joint training as reference upper bound. In terms of Recall rate tested on the last new class group (*i.e.*, *classes (176-200)* and *classes (106-120)*), we find all six incremental learning algorithms and the fine-tuning method (without any knowledge distillation) have similar performance, close to the upper bound, especially for the Stanford-Dogs dataset. However, in terms of Recall on previous tasks, feature correlations used as knowledge can lead to a better-performing performance than other counterparts, closer to the upper bound, which means that our method suffers less forgetting on these preceding tasks. For instance, when tested the final model f_4*(176-200)* on the old *classes (1-100)* of the CUB-200 dataset, our method achieves around 73% of Recall@1, 7% lower than the upper bound (80%), whereas other methods achieve less than 70%.

We have demonstrated that our method can reduce the catastrophic forgetting on the previous tasks effectively. Also, the performance of the new task is essential to evaluate. As the incremental training proceeds, we report the Recall@1 on the new task during each incremental step in Figure 6.9. That is, we record the accuracy of new classes every time these classes are added. The results illustrate the evolution of performance on new classes. Obviously, we observe that all methods have similar Recall evolution and their performance is close to each other, especially for the Stanford-Dogs dataset.

We evaluate the case when more tasks are added sequentially on the CUB-200 dataset. Concretely, the remaining m=100 new classes are divided into 10 groups evenly. We focus on activation regularization algorithms and compare with L2 and MMD loss regularized methods. After the final model f_{10}*(191-200)* is trained at the end of the task sequence (*i.e.*, new *classes (191-200)*), we test this model on the original *classes (1-100)*, which suffer the most severe forgetting. The results are reported in Figure 6.10. Obviously, on the original *classes (1-100)*, correlations distillation with feature estimation method reduces the forgetting on *classes (1-100)* effectively.

6.4.4 Ablation study

a. Efficacy of feature estimation

Feature estimation is introduced in Eq. 6.11 to reduce forgetting in the multi-task scenario. Here, we explore the efficacy of feature estimation. For this purpose, we consider a vanilla correlations distillation only from task (t–1) to task t, *i.e.*, without using the feature estimation. Therefore, the loss for training is $L = \lambda_1 L_{triplet} + \lambda_2 L_{corr}^{(t-1)\rightarrow t}$.

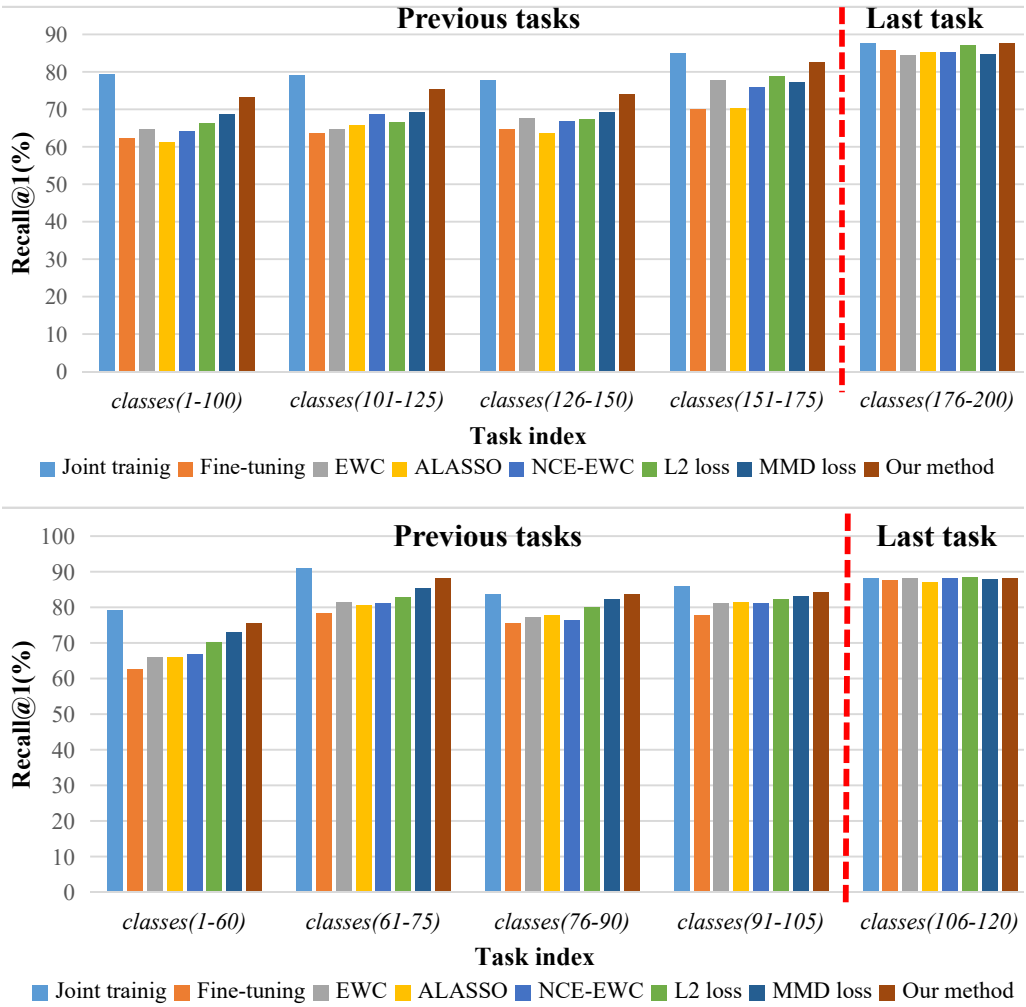


Figure 6.8: The Recall@1 evaluation of each task (class group) at the end of the 4-step incremental learning. For instance, the model f_4*(176-200)* incrementally-trained on 4th new *classes (176-200)* at task $t = 4$ and is tested on all previously seen class groups. (a) Tested on the CUB-200 dataset; (b) Tested on the Stanford-Dogs dataset.

We follow previous experimental protocols and conduct this study on the CUB-200 dataset. We depict the Recall@1 and mAP evolution in Figure 6.11. Note that it is unnecessary to estimate feature drifts when task $t = 1$. When more new classes are added, distilling as knowledge feature correlations like MMD loss and our vanilla distillation method is more effective than L2 loss for reducing performance degradation. Also, vanilla distillation without feature estimation has a higher performance than MMD loss. When feature estimation strategy is used, additional regularization from unsaved models can effectively retain more previously-learned knowledge, thereby leading to less forgetting on the original *classes (1-100)*.

b. Influence of hyper-parameter

We show the efficacy of feature estimation in Figure 6.11. However, it seems that the estimated features in Eq. 6.11 act as augmented components for reducing catas-

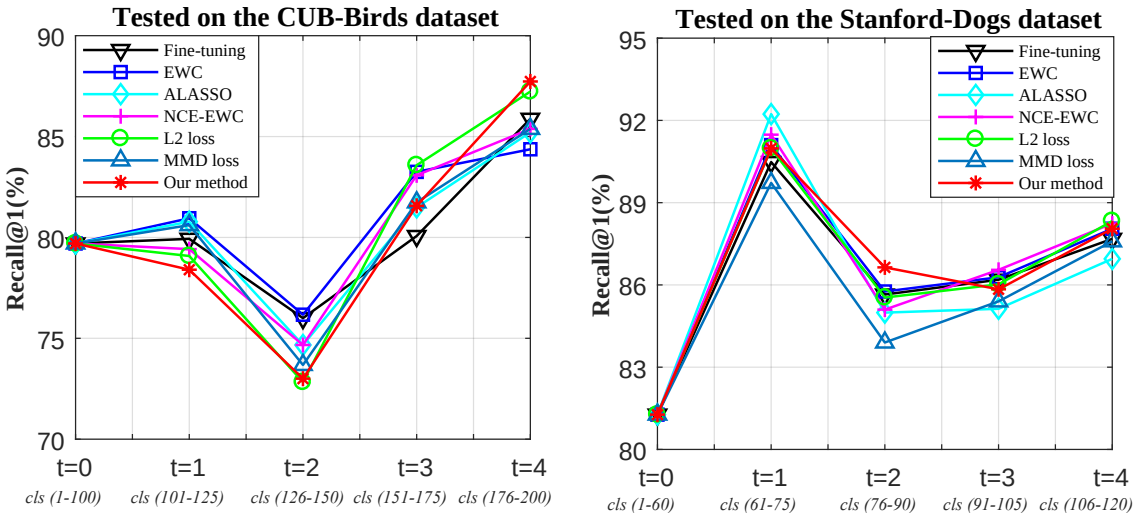


Figure 6.9: The Recall@1 evolution tested on each new incoming class group during incremental learning. "*cls (1-100)*" indicates "*classes (1-100)*".

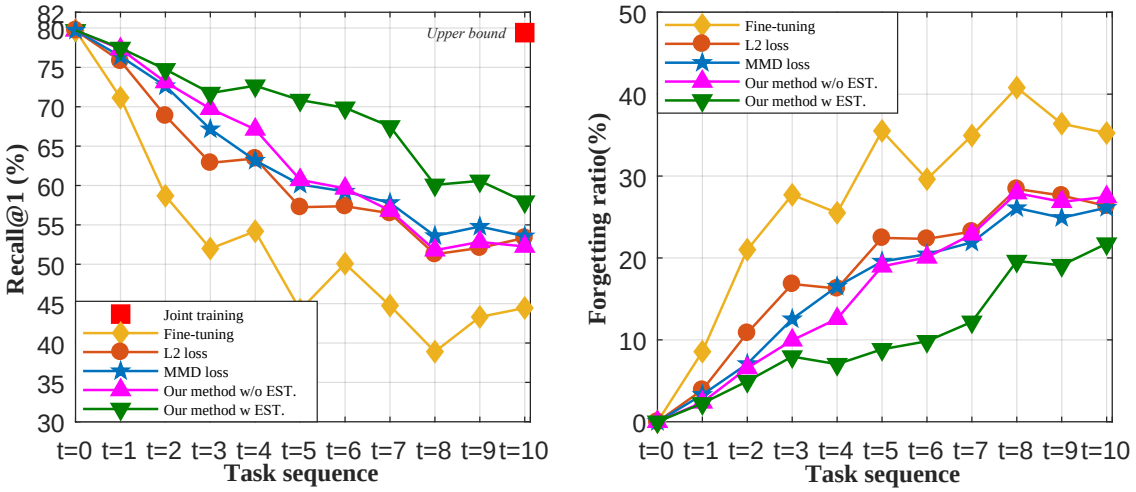


Figure 6.10: 10-task performance comparison on the old *classes (1-100)*. The testing model is trained at the end of 10 tasks sequence on CUB-200. (a) Evolution of Recall@1; (b) Forgetting ratio evaluated on Recall@1.

trophic forgetting. In other words, the forgetting ratio reducing on the old classes might be realized by the hyper-parameter. To this end, we explore the influence of hyper-parameter. Following previous experimental protocols, we consider two-step incremental training on the CUB-200 dataset where only new *classes (101-125)* and *classes (126-150)* are sequentially added. We do not consider task t=1 is because there is no feature estimation in this task. When new *classes (126-150)* are adding at task $t=2$, the deep network is trained, using Eqs. 6.3 and 6.11, under four conditions: **case (a)** without feature estimation, **case (b)** with hyper-parameter augmented, **case (c)** with feature estimation, and **case (d)** with two-model distil-

lation. The case (a) is viewed as a baseline where the correlations are distilled only from the penultimate model f_1*(101-125)* to the to-be-trained model f_2*(126-150)* by using their actual features. The case (d) is a complete method, similar to [235] in which the previous models f_0*(1-100)* and f_1*(101-125)* are both saved for regularizing the training of current task $t = 2$. In contrast, it is unnecessary for our method (case (c)) to save the model f_0*(1-100)*.

The results are reported in Table 6.3. Naturally, the complete method in the case (d) produces an optimal performance on the old classes because all models are available. In terms of the baseline method, due to no distillation regularization, the trained model f_2*(126-150)* has the best performance on the new classes. For instance, its mAP reaches the maximal 52.45%. However, this model degrades performance heavily on the old classes to a minimal mAP (48.09%). In contrast, when the hyper-parameter of the baseline is augmented from λ_2 to $\lambda_2(1 + \frac{1}{(t-1)})$. The trained model f_2*(126-150)* reduces forgetting on the old classes but limits the learning on the new classes. In particular, compared to the baseline method, the mAP of the case (b) on the old *classes (1-100)* reaches a maximal 50.71%, while it has the lowest Recall@1 (75.00%) and mAP (50.87%) on the new *classes (126-150)*. Therefore, Simply increasing the hyper-parameter of the stability term λ_2 in Eq. 6.3 cannot tackle well the stability-plasticity dilemma on the old tasks and new task because no extra knowledge is transferred. By contrast, training by using the feature estimation method can achieve competitive accuracy, taking both the old classes and new classes into account. Specifically, the model trained using the feature estimation method has a similar performance to the "two-model distillation" method on the old classes (76.19% → 76.91% of Recall@1). Meanwhile, the performance on the new classes is close to that of the baseline method (76.33% → 76.83% of Recall@1).

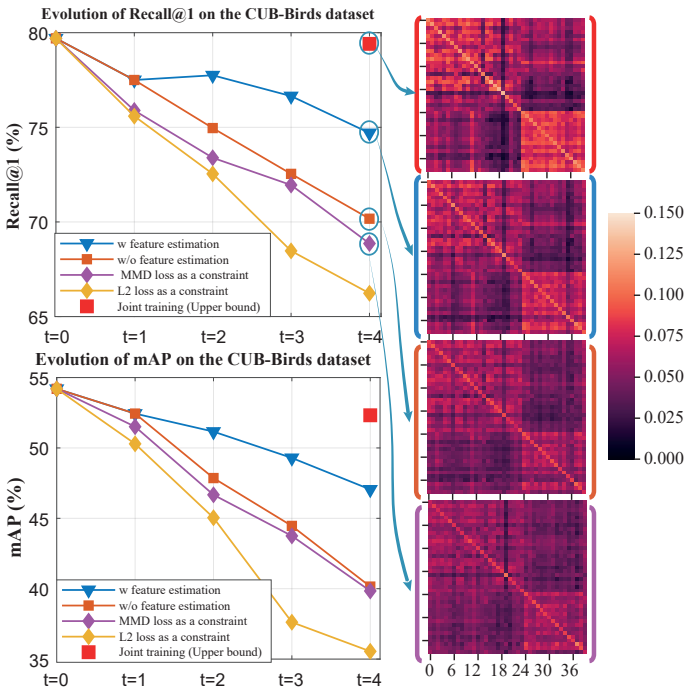


Figure 6.11: Efficacy exploration for (a) Recall@1 and (b) mAP evolution *only* tested on the original *classes (1-100)*. We show the correlation matrices at the end of incremental training. This visualization further indicates that learning with feature estimation makes its performance closer to the upper bound.

Table 6.3: Hyper-parameter analysis (%) on CUB-200 where training task $t = 2$. We set $\lambda_1 = 1$ and $\lambda_2 = 10$. L_{actual} means that the loss term is computed by using actual features, whereas $L_{virtual}$ denotes the one computed by using estimated features.

Configurations		Old *classes (1-100)*		New *classes (126-150)*	
Conditions	The form of loss function $L =$	Recall@1	mAP	Recall@1	mAP
Case (a)	$\lambda_1 L_{triplet} + \lambda_2 \left(L_{actual}^{(t-1)\to t} \right)$	74.75	48.09	76.83	52.45
Case (b)	$\lambda_1 L_{triplet} + \lambda_2 \left(L_{actual}^{(t-1)\to t} + \frac{1}{(t-1)} L_{actual}^{(t-1)\to t} \right)$	76.69	50.71	75.00	50.87
Case (c)	$\lambda_1 L_{triplet} + \lambda_2 \left(L_{actual}^{(t-1)\to t} + \frac{1}{(t-1)} L_{virtual}^{(t-2)\to t} \right)$	76.19	50.45	76.33	51.79
Case (d)	$\lambda_1 L_{triplet} + \lambda_2 \left(L_{actual}^{(t-1)\to t} + L_{actual}^{(t-2)\to t} \right)$	76.61	50.49	76.50	51.92

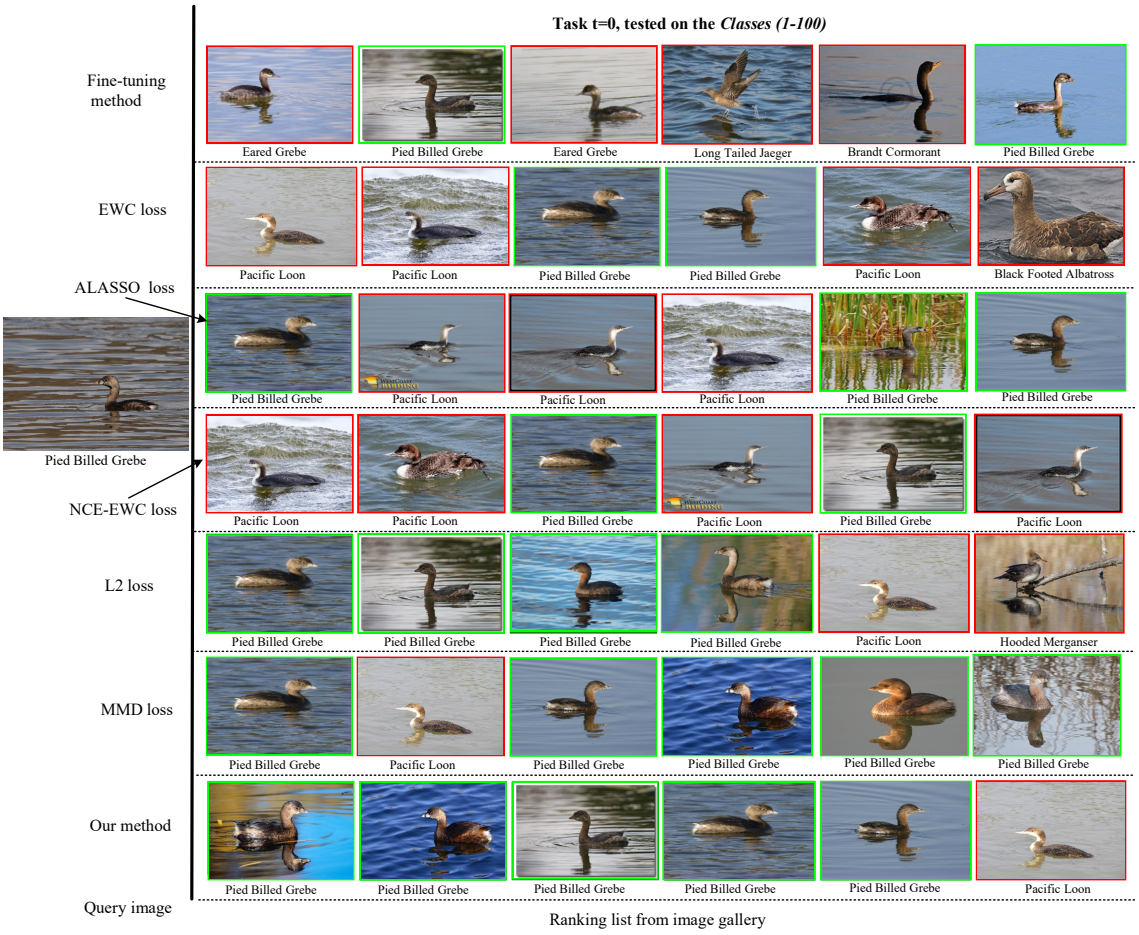


Figure 6.12: Visualization of retrieved images and their class names on the CUB-200 dataset. The Top 6 images tested on *classes (1-100)* are listed from left to right.

6.4.5 Retrieval visualization

We visualize the retrieval results for different methods on 4-tasks sequentially incremental learning on the CUB-200 dataset. *For all methods on different incremental stages, the query image is the same. The red box means an image is retrieved incorrectly, while the green box indicates the retrieved image has the same class label as the query image.* We use the model trained at the end of the 4-step sequentially

incremental training, *i.e.*, the model f_4*(176-200)*, and test this model on the old *classes (1-100)*. Considering the differences among images are subtle, we report the retrieved images and corresponding class names. We select an image from class "*Pied Billed Grebe*" as the query item. This image is difficult to retrieve and is prone to cause forgetting issue because the color of the object in this image is similar to the background, as well as its incomplete appearance. The top 6 retrieved results are shown in Figure 6.12. Overall, all methods can return images with similar scenes. Other incremental algorithms suffer catastrophic forgetting and return more incorrect images. By contrast, our method effectively reduces the forgetting ratio and still returns more correct images of the old tasks after a process of 4-step incremental learning. When the model f_4*(176-200)* are validated on on previous tasks: *classes (101-125)*, *classes (126-150)*, and *classes (151-175)*. Note that *classes (176-200)* are used as the current new classes. The visualizations are depicted in Figure 6.13, Figure 6.14, Figure 6.15, and Figure 6.16, respectively.

6.5 Chapter Conclusions

In this chapter, we explored fine-grained image retrieval in the context of incremental learning, where one-task and multi-task scenarios are validated. To achieve a trade-off performance for old tasks and new tasks, we used new data only and regularized their features extracted from the teacher model and the student model. In terms of multi-task incremental learning, saving all previous models for correlations distillation may cause a great demand in memory storage. We made an attempt to address the issue via a feature estimation method. That is, instead of storing a stream of old models, we saved the accuracy of models to compute the accuracy change during training each task. The semantic correlations of the estimated features, as an additional regularization, further mitigated the catastrophic forgetting ratio on previous tasks. Compared to previous approaches, the advantages of the proposed method were verified by thorough quantitative and qualitative results on two fine-grained datasets. Now, incremental image retrieval methods still need supervisory information. In the future, it is potentially valuable to explore incremental image retrieval in an unsupervised learning manner. Further, the data used in old tasks and new tasks share similar semantic commonalities, it is also interesting to examine for heterogeneous data.

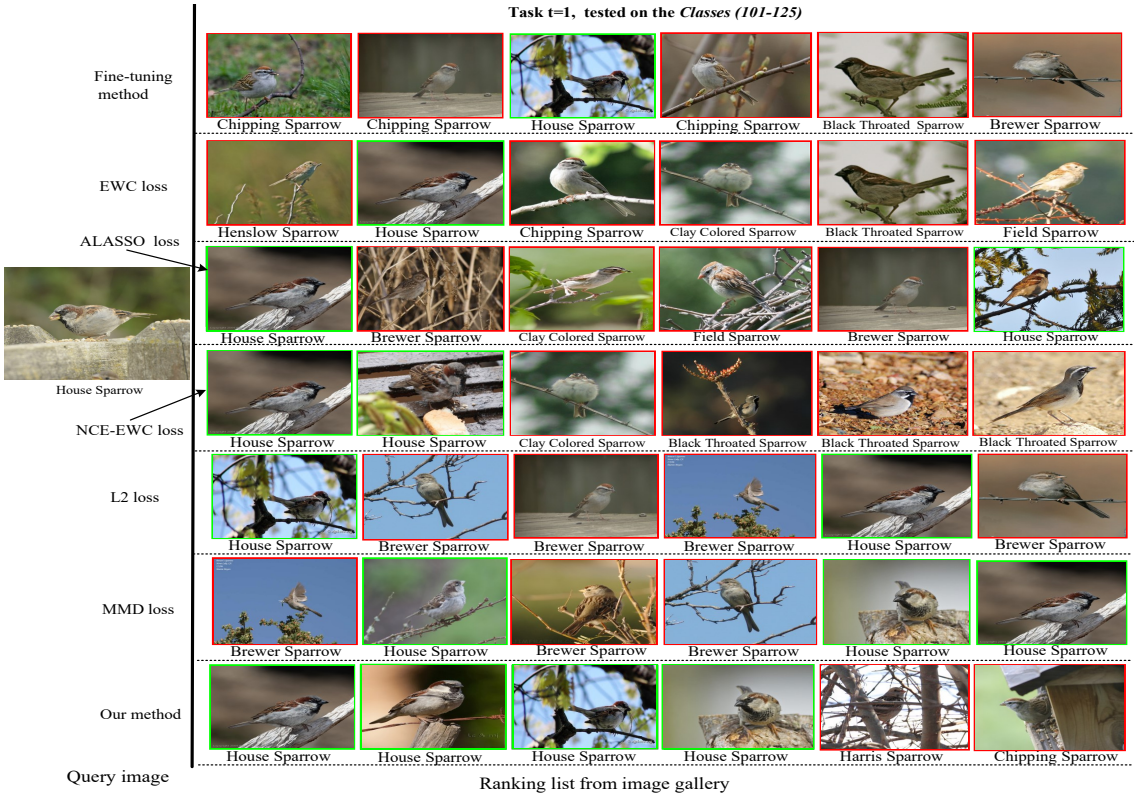


Figure 6.13: The top-6 retrieval results of the model f_4*(176-200)* tested on the previous old *classes (101-125)*.

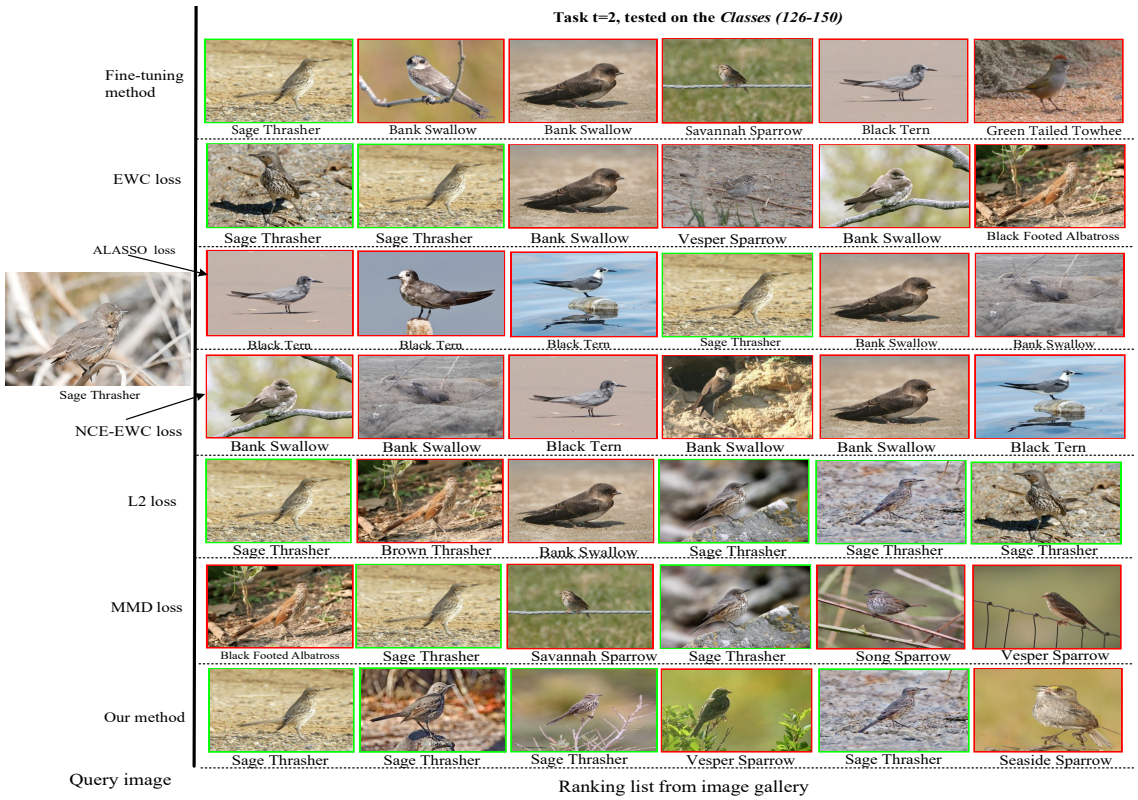


Figure 6.14: The top-6 retrieval results of the model f_4*(176-200)* tested on the previous old *classes (126-150)*.

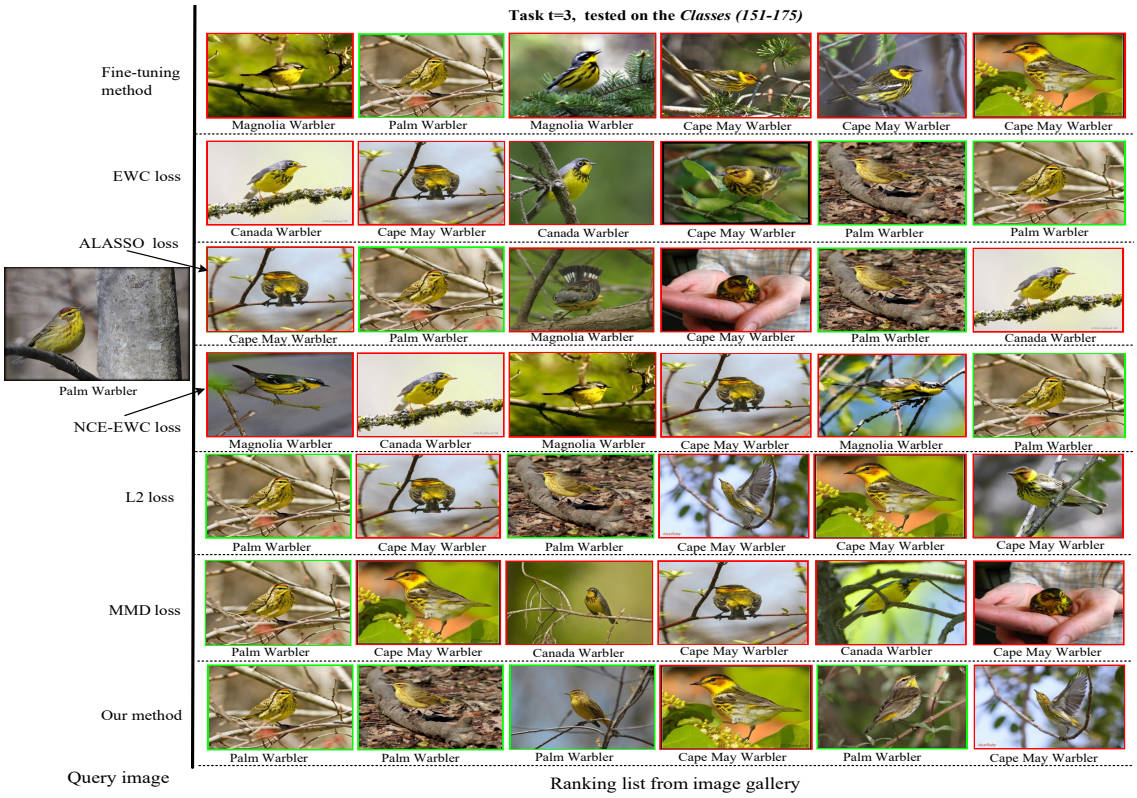


Figure 6.15: The top-6 retrieval results of the model f_4*(176-200)* tested on the previous old *classes (151-175)*.

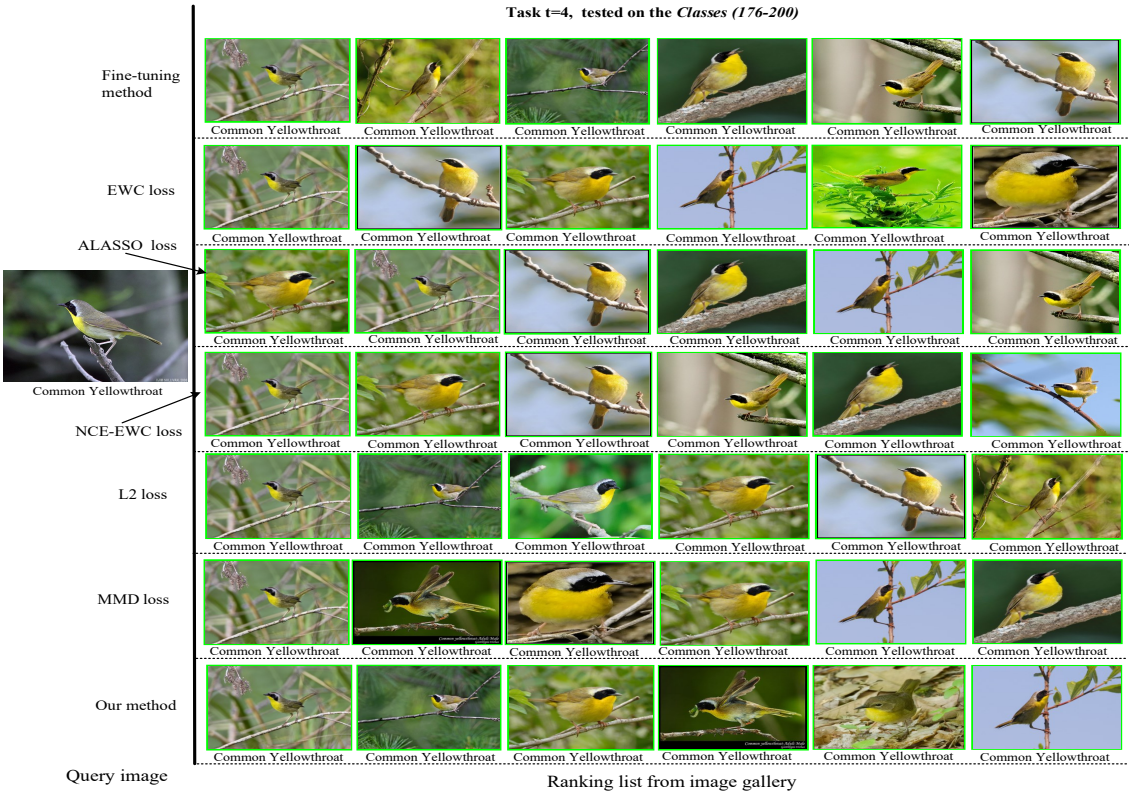


Figure 6.16: The top-6 retrieval results of the model f_4*(176-200)* tested on the current new *classes (176-200)*.

Chapter 7

Lifelong Image Retrieval via Dual Knowledge Distillation

In Chapters 5-6, we explored incremental learning on fine-grained datasets. However, this is still far from realizing the model's continuous retrieval ability because the images in old categories and new categories are similar semantically. Instead, the images in new categories may have different semantic contents (*i.e.* semantic shifts). For the context of incremental learning, the semantic shifts make the problem of minimizing the forgetting ratio more difficult.

In this chapter, we investigate RQ 4, with a goal of gradually transferring acquired knowledge for any new task while minimizing the forgetting ratio on old tasks. To this end, we propose a Dual Knowledge Distillation (DKD) framework consisting of two professional teachers and a self-motivated student. One teacher is trained on previous datasets and then freezes its parameters. This frozen teacher is responsible for transferring previous knowledge. The other teacher is trained jointly with the student by using samples from the new incoming dataset only. This "on the fly" teacher is responsible for learning new knowledge and acts as an assistant model to improve the student's generalization ability. As the incremental learning proceeds, the semantic drifts between the old and new datasets often weaken the effectiveness of knowledge distillation by the frozen teacher. To mitigate this problem, we leverage the stored statistics in the BatchNorm layers of the frozen teacher to generate representative images of the old datasets.

Keywords

Lifelong image retrieval, Dual knowledge distillation, Data generation, BatchNorm statistics

This chapter is based on the following publication:

- Chen, W., Pu, N., Liu, Y. , Lao, M., Wang, W., Bakker, E. M., Liu L., Tuytelaars, T., and Lew, M.S., "Lifelong Image Retrieval via Dual Knowledge Distillation." submitted to Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI) (*under review*), 2021.

7.1 Introduction

Image retrieval have been widely explored in the literature since the emergence of deep learning [130, 131, 132, 226]. Typically, existing retrieval works focus on improving the networks' generalization ability and assume that the target dataset is stationary and fixed. This assumption, however, is infeasible for many real-world scenarios, where the environment is non-stationary. To this end, lifelong learning [251] is proposed to make deep networks learn sequential tasks and adapt to streaming data.

The main challenge for lifelong learning systems is to overcome catastrophic forgetting [252]. Knowledge distillation [214] can be used to reduce forgetting, by transferring the learned information from a trained network (*i.e.* teacher) to a new one (*i.e.* student) [212]. It has been researched for various classification-based tasks, including image classification [213], object detection [217], image generation [216]. However, its efficiency on image retrieval is still less studied due to the challenges below.

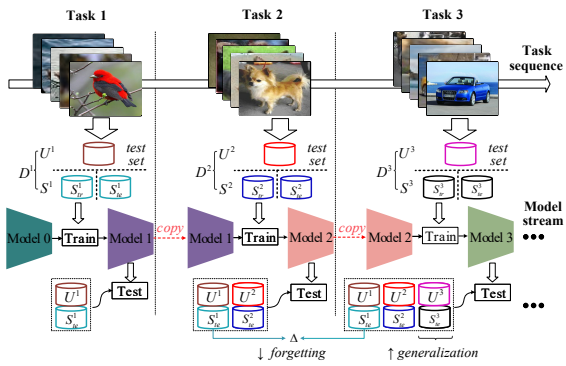


Figure 7.1: Illustration of lifelong image retrieval. A deep model is trained on different sequential datasets $\mathcal{D}^1, \mathcal{D}^2, \mathcal{D}^3, \cdots$. Each dataset is split into a set of seen categories $\mathcal{S}$ and a set of unseen categories $\mathcal{U}$. The semantic difference (*e.g.* birds v.s. cars) results in forgetting when the model is trained on a sequence of task. Thus, the goal is to train the model to minimize the forgetting ratio on the old tasks and simultaneously improve generalization on the new task.

First, a deep model learns to retrieve incrementally on different tasks, and the semantic drifts between the training data lead to tasks that maybe very weakly related, for example the birds, dogs and cars in Figure 7.1. Thus, knowledge distillation cannot effectively prevent the forgetting on streaming data across different tasks. Second, the weak relatedness between tasks results in significant updates of model's parameters when this model learns a new task. Image retrieval is highly sensitive to the matching between features. Thus a small change in the features would have a significant impact on feature matching. The changes in output features make the problem of minimizing forgetting more difficult. Third, conventional knowledge distillation framework pays more attention on preserving the knowledge in the teacher network. This may make it hard to pursue an optimal balance between minimizing the forgetting ratio and improving network's retrieval generalization capacity.

In this chapter, we focus on the three challenges and propose a *Dual Knowledge Distillation* (DKD) framework which includes two professional teachers and a stu-

dent. On the one hand, the first teacher has been trained on previous tasks to transfer old knowledge. To further alleviate the forgetting of the student, we use the statistics stored in the BatchNorm layers of the frozen teacher to generate images used as representatives for the previous datasets. Instead of storing a small budget of exemplars derived from the old data or synthesizing images via training additional generative networks, the representative images can be directly generated from the frozen teacher, without any other operations. On the other hand, the second teacher is trained jointly with the student by using samples from the new task only. This "on-the-fly" teacher acts as an assistant model to improve the student's generalization ability on the new task. Finally, the student can achieve an optimal balance between minimizing the forgetting ratio and improving generalization performance.

7.2 Related Work

Lifelong learning a.k.a. incremental learning, has been explored in image classification [213], object detection [217], image generation [216], and image retrieval [37, 218] *etc*The methods can be broadly divided into three methodologies: network architecture-based [230], memory replay-based [221, 231], and regularization-based methods [213, 227]. Knowledge distillation is one of the regularization-based methods, which can be performed on either the final classifier or the intermediate layers. The key is to minimize the differences between the teacher and the student, which can be characterized by cross-entropy [214], L1 loss [216], L2 loss [253], Gramian matrix [238], and KL-divergence [214]. Multi-teacher knowledge distillation methods have been explored [237]. The ensemble of multiple teachers, *e.g.* by averaging their responses, can provide more powerful prior information for supervising the student. In this chapter, we propose a dual knowledge distillation framework which includes two professional teachers for transferring both old and new knowledge information.

Metric learning has been explored broadly for image retrieval [130, 131, 132, 226]. Given binary indicator information for samples (*i.e.* positive or negative), deep networks learn an embedding space for the features which should be verified as positive pairs or negative pairs [254]. To date, the mainstream methods train deep networks on the seen classes of a fixed dataset and then their generalization performance are validated on the unseen classes of this dataset. Therefore, metric learning for image retrieval focuses on the forward transfer [230], *i.e.* transferring a positive influence to improve the performance on future unseen data. Nevertheless, these methods do not consider the negative backward transfer issue (*i.e.* catastrophic forgetting). Therefore, we explore lifelong image retrieval, with the goal to reduce forgetting and simultaneously improve generalization ability.

BatchNorm statistics utilization. The statistics stored in the BatchNorm layers of a pre-trained model are used for data-free knowledge distillation [255] and data-free model compression [256]. These statistics are relevant to the statistical characteristics of the datasets trained in the past. They have been used as a reference to generate images. For instance, Yin *et al.* [255] introduced Adaptive DeepInversion (ADI) which is a feature map regularizer based on BatchNorm statistics that enables image synthesis from random noise. The generated images have similar semantics to the images of ImageNet. The images generated in [255] depend on optimizing the gradients computed from cross-entropy loss based on the given class labels. This is not directly applicable to lifelong image retrieval because (1) the order of given class labels may affect the softmax-based probabilities of a classifier as the tasks are added sequentially; (2) lifelong image retrieval tasks do not depend on softmax-based probabilities to perform. Instead, we apply a clustering loss to generate images.

7.3 The Lifelong Image Retrieval Problem

Preliminary. To perform image retrieval, a dataset $\mathcal{D}$ is split into a training set $\mathcal{D}_{tr}$ and a testing set $\mathcal{D}_{te}$. A deep network $f(\cdot, \boldsymbol{\theta})$ is trained on $\mathcal{D}_{tr}$ to learn representations $\boldsymbol{F} = f(X, \boldsymbol{\theta})$ by using a certain objective function. To date, ranking loss has been widely used as a constraint to train the network f. Taking the triplet loss as an example, each ground-truth label in $\mathcal{D}_{tr}$ is used to mine a positive x_p, a hard negative x_n, and an anchor image x_a. The network f is trained to learn a feature space, where the distance between x_n and x_a denoted by $D(x_a, x_n) = ||f(x_a; \boldsymbol{\theta}) - f(x_n; \boldsymbol{\theta})||_2^2$ is pushed away by a margin $\delta > 0$ from $D(x_a, x_p)$:

$$L_{triplet}(x_a, x_p, x_n) = \max\Big(\delta + D(x_a, x_p) - D(x_a, x_n), 0\Big) \tag{7.1}$$

Problem definition. We use the triplet loss as a basic constraint to train a model to perform tasks incrementally. The flowchart is illustrated in Figure 7.1. Each task t corresponds to the training of a whole dataset $\mathcal{D}^t$ (*e.g.* birds). During the tth task, dataset $\mathcal{D}^t$ is split into a set of seen categories $\mathcal{S}^t$ and a set of unseen categories $\mathcal{U}^t$. For the seen part, $\mathcal{S}^t$ includes n_s categories, *i.e.* $\mathcal{S}^t = \{(X^c, y^c) | c = 1, 2, \cdots, n_s\}$, each class c includes a different amount of images $|X^c|$ sharing the same label y^c. The $\mathcal{S}^t$ part is further split into a training set and a testing set. Likewise, the unseen part $\mathcal{U}^t$ includes n_u categories, all of which are used to evaluate the model's generalization ability, similar to the general practice in metric learning for image retrieval. For lifelong image retrieval, suppose a deep model has been trained sequentially on the training sets $\mathcal{S}^1, \mathcal{S}^2, \cdots, \mathcal{S}^t$ (current task t). On the one hand, it is required that the trained model is able to minimize the forgetting ratios on the previous tasks $\mathcal{S}^1, \mathcal{S}^2, \cdots, \mathcal{S}^{t-1}$ and $\mathcal{U}^1, \mathcal{U}^2, \cdots, \mathcal{U}^{t-1}$, thereby retaining its retrieval capacity on the previous datasets $\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^{t-1}$. On the other hand, it is required that the

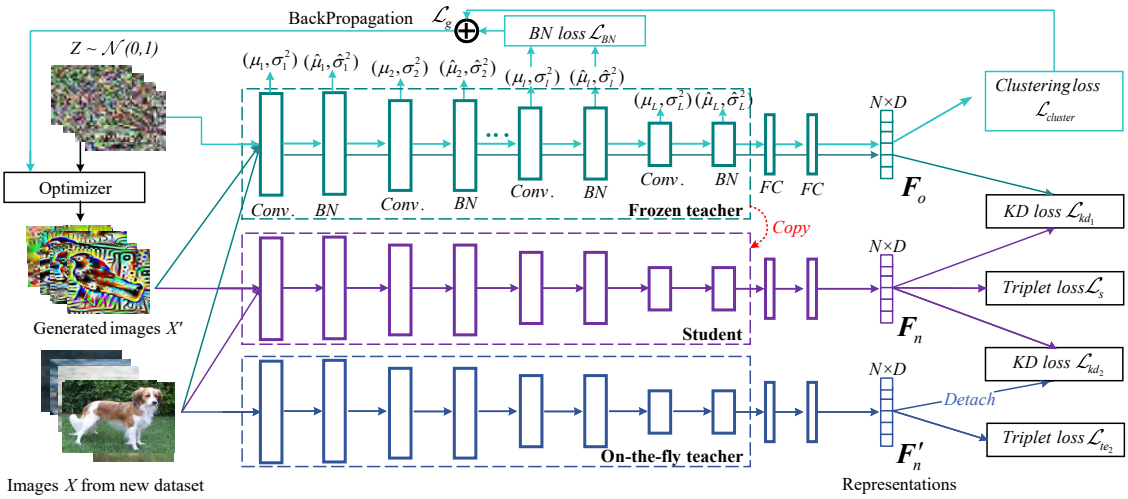


Figure 7.2: The dual knowledge distillation (DKD) framework. The stored statistics in the BatchNorm layers of the frozen teacher are used to generate representative images, optimized by the L_g. The on-the-fly teacher is initialized its parameters differently from the frozen teacher and trained jointly with the student by using L_{te_2}. For clarity, the ReLU activation function and pooling layers are not depicted.

trained model achieves good accuracy on the seen part $\mathcal{S}^t$ and, more importantly, generalizes well on the unseen part $\mathcal{U}^t$ of current dataset $\mathcal{D}^t$.

7.4 Dual Knowledge Distillation

To minimize the forgetting ratio and simultaneously improve generalization performance, we propose a dual knowledge distillation (DKD) framework which includes two teachers and a student, as shown in Figure 7.2. In the following, we will introduce each component in more detail.

7.4.1 Knowledge distillation by frozen teacher

Prior to training task t, a teacher model has been trained on the previous task $(t-1)$ and has its parameters fixed. Training the student on the new task t leads to a negative backward transfer which may degrade the performance of preceding tasks [230]. Knowledge distillation by using the frozen teacher $f_{te_1}^{t-1}$ can prevent this degradation. As shown in Figure 7.2, knowledge distillation by using the frozen teacher is performed on the embedded D-dimension features from the fully-connected layers, formulated as $\boldsymbol{F}_o = f_{te_1}^{t-1}(X^c, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{N\times D}$, where N is the size of a mini-batch. Likewise, the feature representations from the student f_s^t are $\boldsymbol{F}_n = f_s^t(X^c, \boldsymbol{\theta}_s^t) \in \mathbb{R}^{N\times D}$. As suggested in [245, 257, 258], semantically similar inputs produce similar patterns on features extracted from the frozen teacher and the student. Therefore, we adopt the Gram matrix with a kernel function to measure the feature correlations:

$$G_o^{(i,j)} = \mathcal{K}(F_o^i, F_o^j);\ G_n^{(i,j)} = \mathcal{K}(F_n^i, F_n^j) \tag{7.2}$$

$\mathcal{K}(\cdot)$ refers to inner product, *i.e.*, $\mathcal{K}(F^i, F^j) = <F^i, F^j>$. Each entry (i, j) in $\boldsymbol{G} \in \mathbb{R}^{N\times N}$ represents the correlations of the same activation $(i = j)$ or these between different activations $(i \neq j)$. We use KL-divergence to measure the difference between $\boldsymbol{G}_o$ and $\boldsymbol{G}_n$, normalized by a Softmax function $\sigma(\cdot)$. Thus, the knowledge distillation loss by the frozen teacher $f_{te_1}^{t-1}$ is formulated as L_{kd_1}, weighted by a factor λ_{kd_1}:

$$L_{kd_1} = \lambda_{kd_1} \sum^{N} KL\Big(\sigma(\boldsymbol{G}_o), \sigma(\boldsymbol{G}_n)\Big) \tag{7.3}$$

7.4.2 Representative data generation

When the student learns task t, the performance degradation of preceding tasks is prevented by using the KL-divergence in Eq. 7.3. However, when the student is trained incrementally on the data with large semantic drifts (*e.g.* birds and cars in Figure 7.1), L_{kd_1} cannot effectively prevent the degradation by transferring more previously learned information. To overcome this problem, we use the stored statistics in BatchNorm layers to generate samples as representatives for the previous tasks. Representative data generation is performed by the frozen teacher itself, instead of selecting exemplars from these already-trained datasets.

Suppose the frozen teacher includes L convolutional layers, each of which is followed by a BatchNorm layer, as shown in Figure 7.2. Each BatchNorm layer l includes channel-wise running means $\hat{\mu}_l$ and running variances $\hat{\sigma}_l^2$. Prior to training the student, a batch of Gaussian noise Z with random class labels Y' are fed into the teacher. Outputs of each convolutional layer l of the teacher are used to compute the batch means μ_l and batch variances σ_l^2. Similar to [255], we define a BN loss L_{BN} to measure the difference between the stored statistics and the current statistics of Z:

$$L_{BN} = \lambda_{BN} \textstyle\sum_{l=1}^{L}\Big(\|\mu_l(Z) - \hat{\mu}_l\|_2^2 + \|\sigma_l^2(Z) - \hat{\sigma}_l^2\|_2^2\Big) \tag{7.4}$$

Different from ADI in [255] which is limited only from the classification networks. We apply a K-means clustering loss $L_{cluster}$, together with L_{BN} to optimize Z. Given a mini-batch of N noise tensors with K classes, containing P tensors of each given class, the mean M_k for a class $k \in K$ is defined as $M_k = \frac{1}{P}\sum_{p=1}^{P} f_{te_1}^{t-1}(z_{k_p}, \boldsymbol{\theta}_{te_1}^{t-1})$, where z_{k_p} is a sample from the tensors Z. The number of clusters is set to the number of classes in tensors Z (*i.e.* K classes). We cluster features of Z via computing intra-class and inter-class distances. Specifically, for a given class $k \in K$, a set of intra-class distances d_k^{intra} is formulated as $\{||f_{te_1}^{t-1}(z_{k_p}, \boldsymbol{\theta}_{te_1}^{t-1}) - M_k||_2|\}$, where $p = 1, 2, ..., P$ and the number of elements in d_k^{intra} is equal to P. Likewise, a set of

inter-class distances d_k^{inter} is computed according to all other $(N-P)$ samples from k'_p classes ($k'_p \in K$ and $k'_p \neq k$). Clustering all the elements in d_k^{intra} and d_k^{inter} leads to a low training efficiency. Instead, we mine the hardest samples in these distance sets. For d_k^{intra}, we mine the sample that lies farthest from its class mean M_k. For d_k^{inter}, we mine the sample that lies closest from the considered class mean M_k. For all K classes, we use a clustering loss $L_{cluster}$ to regularize the inter-class variations to become larger than the intra-class variations by a margin $\Delta > 0$:

$$L_{cluster} = \lambda_{cluster} \sum_{k=1}^{K} \max\Big(\Delta + \max_{P} d_k^{intra} - \min_{N-P} d_k^{inter}, 0\Big) \tag{7.5}$$

Afterwards, the loss $L_g = L_{BN} + L_{cluster}$ is used to optimize Z based on the frozen teacher $f_{te_1}^{t-1}(\cdot, \boldsymbol{\theta}_{te_1}^{t-1})$ to generate representative images X' of the previous task $(t-1)$, *i.e.* $X' \leftarrow \underset{z \in Z}{\operatorname{argmin}} \sum(L_g; \boldsymbol{\theta}_{te_1}^{t-1})$. Images X' and class labels Y' can be used to build a mixed dataset $X_{mix} = X \cup X'$. X belongs to the origin training set in D^t. The mixed labels are $Y_{mix} = Y \cup Y'$. In this case, the mixed data are fed into the frozen teacher $f_{te_1}^{t-1}$ to transfer richer previous knowledge to the student.

7.4.3 Self-motivated learning on the mixed data

At the start of task t, the parameters of the student are copied from the frozen teacher, as shown in Figure 7.1. The self-motivated learning for the student is important for guaranteeing the performance on the current task t, as can be seen from the results for *Case 4* in Table 7.6. Consistent to the training scheme for the frozen teacher, we employ the triplet loss in a similar form as Eq. 7.1 to train the student.

$$L_s = \lambda_s \sum^{N} L_{triplet}\Big(f_s^t(x'_a), f_s^t(x'_p), f_s^t(x'_n)\Big) \tag{7.6}$$

Note that the anchor, positive, and negative images (x'_a, x'_p, x'_n) are from the mixed dataset X_{mix} according to the mixed labels Y_{mix} in each training session.

7.4.4 Auxiliary distillation by on-the-fly teacher

During training, the student needs to learn new information and simultaneously protect previous knowledge. However, knowledge distillation from the mixed data using the frozen teacher is a strong regularization by the time it reaches the student, making the student be prone to remembering previous knowledge but having lower generalization on the new task t, as demonstrated by *Case 2* in Table 7.6. As a result, an optimal balance between reducing forgetting and improving generalization is hard to achieve. Therefore, we propose a second teacher $f_{te_2}^t$ which is trained together with the student. Its parameters $\boldsymbol{\theta}_{te_2}^t$ are initialized differently from these

of the frozen teacher and the student. This teacher is constrained by a triplet loss L_{te_2}:

$$L_{te_2} = \lambda_{te_2} \sum^{N} L_{triplet}\Big(f^t_{te_2}(x_a), f^t_{te_2}(x_p), f^t_{te_2}(x_n)\Big) \tag{7.7}$$

For L_{te_2}, the training images (x_a, x_p, x_n) are mined only from $S^t = \{(X^c, y^c)|c = 1, 2, \cdots, n_s\}$ of the dataset D^t, rather than the mixed data X_{mix}, see Figure 7.2. The on-the-fly teacher is designed to transfer new information to the student to improve its generalization ability. Thus, an auxiliary knowledge distillation loss L_{kd_2} is defined as:

$$\begin{aligned} &L_{kd_2} = \lambda_{kd_2} \sum^{N} KL\Big(\sigma(\boldsymbol{G}'_n), \sigma(\boldsymbol{G}_n)\Big) \\ &\text{where } G'_n = \mathcal{K}(F'_n, F'_n),\ F'_n = f^t_{te_2}(x, \boldsymbol{\theta}^t_{te_2}); \\ &G_n = \mathcal{K}(F_n, F_n),\ F_n = f^t_s(x, \boldsymbol{\theta}^t_s); x \in X \end{aligned} \tag{7.8}$$

Note that during training the gradients computed from L_{kd_2} are *detached* for the on-the-fly teacher. This operation can guarantee the on-the-fly teacher to be fully dedicated to capturing new information from the new dataset $\mathcal{D}^t$.

Full objective. When training with dataset $\mathcal{D}^t$ on task t, together with the generated images, the DKD framework is running by using the full objective function:

$$L = \lambda_s L_s + \lambda_{kd_1} L_{kd_1} + \lambda_{kd_2} L_{kd_2} + \lambda_{te_2} L_{te_2} \tag{7.9}$$

7.5 Experiments

7.5.1 Dataset splits

Our experimental methodology involves using sequences of two tasks and sequences of three tasks in a roughly similar way as the recent lifelong learning research [259]. We perform experiments on three datasets: Caltech-UCSD Birds (CUB-200) [224], Stanford-Dogs [223], and Stanford-Cars [260].

- *CUB-200* includes 11,788 images of 200 classes. We select 150 classes (8,822 images) as the seen set $\mathcal{S}$ and use the remaining 50 classes as unseen set $\mathcal{U}$ (2,966 images). For the seen set, we select ∼60% of each class for training (5,274 images), while the remaining ∼40% (3,548 images) are used to evaluate the forgetting ratio.

- *Stanford-Dogs* includes 20,580 images of 120 classes. We select 100 classes (17,028 images) as the seen set and use the remaining 20 classes as unseen set $\mathcal{U}$ (3,552 images). For the seen set, we select ∼80% of each class for training (13,063 images), while the remaining ∼20% (3,965 images) are for testing.

- *Stanford-Cars* includes 16,185 images of 196 classes. We select 160 classes (10,038 images) as the seen set and use the remaining 36 classes as unseen set $\mathcal{U}$ (3,040 images). For the seen set, we select ∼80% images of each class for training (10,038 images), while the remaining ∼20% (3,107 images) are used at test.

7.5.2 Training details

We utilize the pre-trained Google Inception with BatchNorm as a backbone net. The on-the-fly teacher is always initialized from the pre-stored parameters learned from ImageNet before training each task. Following the practice in [131, 226], the final retrieval features are 512-D. The model is trained for 1500 epochs on the first dataset to get the initial frozen teacher. The training is constrained by the triplet loss with a margin $\delta = 0.5$ as given in Eq. 7.1, optimized by the Adam optimizer with a learning rate of 1×10^{-6} and a batch size of 32, The fully-connected layers for dimension reducing are updated with a learning rate of 1×10^{-5}. Representative images are generated by using Eqs. 7.4 and 7.5 where factors λ_{BN} and $\lambda_{cluster}$ are set to 0.01 and 0.1, respectively. Δ in Eq. 7.5 is set to 1.0. The image generation process is optimized by an additional Adam optimizer with a learning rate of 0.5. The factors λ_s, λ_{te_2}, λ_{kd_1}, and λ_{kd_2} in Eq. 7.9 are set to 1, 1, 80, 20, respectively. We include the main steps of the Dual Knowledge Distillation (DKD) framework in Algorithm 2. Before training each task, the student initializes its parameters from the frozen teacher. Differently, the on-the-fly teacher is always initialized from the pre-stored parameters of Google Inception learned from the ImageNet. In addition, its fully-connected layers are initialized randomly. Image generation process is performed prior to training the student model. The whole framework is trained in an end-to-end manner.

7.5.3 Performance evaluation

Baseline. To the best of our knowledge, there is no prior work for lifelong image retrieval performed on different datasets. We build the sequential fine-tuning (SFT) method as a baseline, which is performed by using a triplet loss as defined in Eq 7.1. We compare 3 knowledge distillation methods, including L_1 loss [216], L_2 loss [217], and maximum mean discrepancy loss (L_{mmd} in short) [37]. We claim the work of incremental fine-grained image retrieval [37] is less challenging than ours because the new data and old are from the same dataset in [37]. Similar to [259], we use the joint training on the training sets of 3 datasets as the *upper-bound* reference for all compared methods.

Metrics. We evaluate the performance of seen set s and that of unseen set u by using the standard performance metric *Recall@K* (*i.e.* *R@K*). The evaluation for u is the same as the one widely explored in deep metric learning [130, 131, 132, 226] which aims at demonstrating the generalization ability. The evaluation for s aims

Algorithm 2: Dual Knowledge Distillation (DKD) framework

Input:
Frozen teacher $f_{te_1}^{t-1}(\cdot, \boldsymbol{\theta}_{te_1}^{t-1})$ has been trained on the previous task $t-1$;
New training images $X \in \mathbb{R}^{N\times H\times W\times 3}$ and labels $Y \in \mathbb{R}^{N\times 1}$ on the training set of $\mathcal{S}^t$ on the current dataset $\mathcal{D}^t$;
Initialization:
$\boldsymbol{\theta}_s^t = \boldsymbol{\theta}_{te_1}^{t-1}$ *//Copied the frozen teacher as the initial student;*
$\boldsymbol{\theta}_{te_2}^t \leftarrow$ Google Inception *//Initialize on-the-fly teacher;*
Random noise tensor $Z \in \mathbb{R}^{N\times H\times W\times 3}$;
Random labels $Y' \in \mathbb{R}^{N\times 1}$ for input noise Z *//Include K classes in total;*
Iterations *Iter* of image generation; Training epochs *Epoch*; Mini-batch size N;
Optimizer with a learning rate lr_1;
Training:
For $iter = 0$ to *Iter*
$\boldsymbol{F}(Z) = f_{te_1}^{t-1}(Z, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{N\times D}$ *//Features to calculate cluster means, inter-class distance sets, and intra-class distance sets;*
$L_{BN} = \sum_{l=1}^{L}\Big(\|\mu_l(Z) - \hat{\mu}_l\|_2^2 + \big\|\sigma_l^2(Z) - \hat{\sigma}_l^2\big\|_2^2\Big)$ *//BN loss in Eq. 7.4;*
$L_{cluster} = \sum_{k=1}^{K}\max\Big(\Delta + \max_{P} d_k^{intra} - \min_{N-P} d_k^{inter}, 0\Big)$ *//Clustering loss in Eq. 7.5;*
$X' \leftarrow \underset{Z}{\operatorname{argmin}} \sum\Big((L_{BN} + L_{cluster}); \boldsymbol{\theta}_{te_1}^{t-1}\Big)$ *//Using the optimizer with lr_1;*
End for
For $epoch = 0$ to *Epoch*
$X_{mix} = X \cup X'$, $Y_{mix} = Y \cup Y'$ *//Build a mixed dataset via data concatenation;*
$\boldsymbol{F}_o = f_{te_1}^{t-1}(X_{mix}, \boldsymbol{\theta}_{te_1}^{t-1}) \in \mathbb{R}^{2N\times D}$ *//$2N \times D$-dim features from the frozen teacher;*
$\boldsymbol{F}_n = f_s^t(X_{mix}, \boldsymbol{\theta}_s^t) \in \mathbb{R}^{2N\times D}$ *//$2N \times D$-dim features from the student;*
$\boldsymbol{F}_n' = f_{te_2}^t(X, \boldsymbol{\theta}_{te_2}^t) \in \mathbb{R}^{N\times D}$ *//$N \times D$-dim features from the on-the-fly teacher;*
$L_{kd_1} = \mathrm{KL}(\boldsymbol{F}_o, \boldsymbol{F}_n)$ *//Knowledge distillation from the frozen teacher in Eq. 7.3;*
$L_s = \mathrm{Triplet}(\boldsymbol{F}_n, Y_{mix})$ *//Triplet loss from the student in Eq. 7.6;*
$L_{te_2} = \mathrm{Triplet}(\boldsymbol{F}_n', Y)$ *//Triplet loss from the on-the-fly teacher in Eq. 7.7;*
$L_{kd_2} = \mathrm{KL}(\boldsymbol{F}_n', \{\boldsymbol{F}_n\}_{n=1,\ldots,N})$ *//Knowledge distillation in Eq. 7.8;*
$L = L_s + L_{kd_1} + L_{kd_2} + L_{te_2}$ *//Weighted full loss function in Eq. 7.9;*
End for
Output: The optimized student model $f_s^t(\cdot, \boldsymbol{\theta}_s^t)$.

to analyze the forgetting ratio of a considered model. Similar to [261], we use the harmonic mean H of s and u to evaluate the trained model, which the most important metrics for each task.

$$H = \frac{2 \times s \times u}{s + u} \tag{7.10}$$

Results. We consider the two-task scenario and three-task scenario. For the two-task scenario, we use CUB-200 as the first task, and consider the task sequences: CUB-200 → Stanford-Dogs and CUB-200 → Stanford-Cars. The results are reported in Tables 7.1 and 7.2. For the three-task scenario, we randomly select a task sequence starting with CUB-200: CUB-200 → Stanford-Dogs → Stanford-Cars. The results

are reported in Table 7.3. For clarity, we report the Recall@1 results.

(1) Two-task evaluation. As shown in Tables 7.1 and 7.2, three experimental comparisons are reported. Compared to the reference, fine-tuning on the Stanford-Dogs and Stanford-Cars achieves a Recall@1 of 78.0% and 77.5% of H on the second task, respectively, while fine-tuning suffers from forgetting on the first task. If "one-teacher" knowledge distillation methods are performed, the student suffers less from forgetting. However, the improvements on the first task are limited due to the semantic drifts. When BatchNorm statistics are used to address this limitation, we observe that the students regularized by different methods are both prone to remembering the first task but degrading their generalization ability on the second task. This is caused by the strong regularization from the frozen teacher, together with the representative images. If the on-the-fly teacher is used (*i.e.* "DKD + BN statistics"), the generalization performance on the second task is improved or even surpass that from the baseline. For instance, on sequence "CUB-200 → Stanford-Dogs" in Table 7.1, when knowledge distillation in the DKD framework is realized by using KL-divergence in Eqs. 7.3 and 7.8, the overall Recall@1 reaches to 80.0%, higher than the 78.0% of the baseline. This demonstrates the efficiency of the auxiliary distillation. At the same time, the student suffers from the minimal degradation on the first task, with a Recall@1 of 67.0%, compared to the 68.7% of the reference. Likewise, on sequence "CUB-200 → Stanford-Cars" in Table 7.2, the student has a Recall@1 of 60.7% compared to 67.7% of the reference. This larger difference is caused by the different distributions between training data of Stanford-Dogs and that of Stanford-Cars.

(2) Three-task evaluation. When three tasks are performed incrementally, the student trained on the final task is tested on the previous two datasets. The results are reported in Table 7.3. Specifically, the generalization performance of the DKD framework on the last task (*i.e.* on Stanford-Cars) is close to or even surpasses the reference performance of joint training (*i.e.* 78.1% and 77.8%). Compared to the two-task scenario, training on the sequence of three tasks leads to more forgetting on the preceding tasks due to the accumulated semantic drifts, especially for the first task. We compare the forgetting ratios of the compared methods on CUB-200. As depicted in Figure 7.3, the initial model is converged at 1500 training epochs on CUB-200, with Recall@1=74.8% on seen set and Recall@1=61.6% on unseen set. We observe that the SFT method degrades performance significantly. Training on the sequence of three tasks also causes forgetting on the unseen set, as shown in Figure 7.3(b). In comparison, the proposed DKD reduces the degradation greatly and is closer to the upper-bound reference.

(3) Evaluation of the on-the-fly teacher. Due to the gradients detach operation, the on-the-fly teacher learns the new task, only being regularized by the term L_{te_2} in Eq. 7.7. We follow the setup of the two-task scenario in Table 7.1, and

Table 7.1: Recall@K (K=1) comparison (%) of s and u for the sequence "CUB-200 → Stanford-Dogs". "KD" represents that one frozen teacher is used for knowledge distillation only. For all cases, the student is regularized by triplet loss only. "KL-divergence" denotes that the knowledge is transferred by using Eq. 7.3. The best balanced results are highlighted in boldface.

		CUB-200 → Stanford-Dogs					
		Test on CUB-200			Test on Stanford-Dogs		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	FT [226]	56.0	47.5	51.4	72.2	84.9	78.0
KD	L_1 loss [216]	52.1	47.4	49.6	71.1	78.7	74.7
	L_{mmd} loss [37]	62.3	52.2	56.8	73.3	85.3	78.9
	L_2 loss [217]	60.5	49.9	54.7	73.7	85.0	78.9
	KL-divergence	62.2	52.1	56.7	73.6	85.0	78.9
KD + BN statistics	L_1 loss [216]	72.0	60.7	65.9	49.8	76.8	60.4
	L_{mmd} loss [37]	73.1	61.7	66.9	49.7	76.3	60.2
	L_2 loss [217]	72.5	62.3	67.0	49.4	75.5	59.7
	KL-divergence	73.5	63.8	68.3	60.0	80.3	68.7
DKD + BN statistics	L_1 loss [216]	64.1	53.3	58.2	74.3	84.8	79.2
	L_{mmd} loss [37]	68.6	60.1	64.1	73.8	85.9	79.4
	L_2 loss [217]	71.7	61.1	66.0	72.1	85.2	78.1
	KL-divergence	**72.0**	**62.7**	**67.0**	**74.4**	**86.5**	**80.0**
Reference	Joint training	74.1	64.1	68.7	74.5	86.7	80.1

Table 7.2: Recall@K (K=1) comparison (%) of s and u for the sequence "CUB-200 → Stanford-Cars". "KD" represents that one frozen teacher is used for knowledge distillation only. For all cases, the student is regularized by triplet loss only. "KL-divergence" denotes that the knowledge is transferred by using Eq. 7.3. The best balanced results are highlighted in boldface.

		CUB-200 → Stanford-Cars					
		Test on CUB-200			Test on Stanford-Cars		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	FT [226]	41.8	38.4	40.0	74.9	80.2	77.5
KD	L_1 loss [216]	43.9	37.1	40.2	72.6	79.2	75.8
	L_{mmd} loss [37]	46.4	39.2	42.5	75.4	79.0	77.2
	L_2 loss [217]	44.5	38.4	41.2	74.7	80.2	77.4
	KL-divergence	45.0	40.5	42.6	74.3	80.6	77.3
KD + BN statistics	L_1 loss [216]	58.7	50.8	54.5	68.4	75.4	71.7
	L_{mmd} loss [37]	64.5	57.2	60.6	64.3	73.6	68.6
	L_2 loss [217]	63.4	56.0	59.5	69.9	76.4	73.0
	KL-divergence	64.5	57.1	60.6	69.8	78.5	73.9
DKD + BN statistics	L_1 loss [216]	54.9	45.4	49.7	73.3	80.6	76.8
	L_{mmd} loss [37]	52.0	**63.8**	57.3	72.7	79.5	76.0
	L_2 loss [217]	57.2	49.9	53.3	74.1	80.4	77.1
	KL-divergence	**64.6**	57.3	**60.7**	**74.6**	**83.5**	**78.8**
Reference	Joint training	72.1	63.8	67.7	77.5	82.2	79.8

report the performance of the on-the-fly teacher under the training sequence: CUB-200 → Stanford-Dogs. Since this teacher is specific for transferring newly-learned information of a new dataset, we only report its performance on the second task (*i.e.*

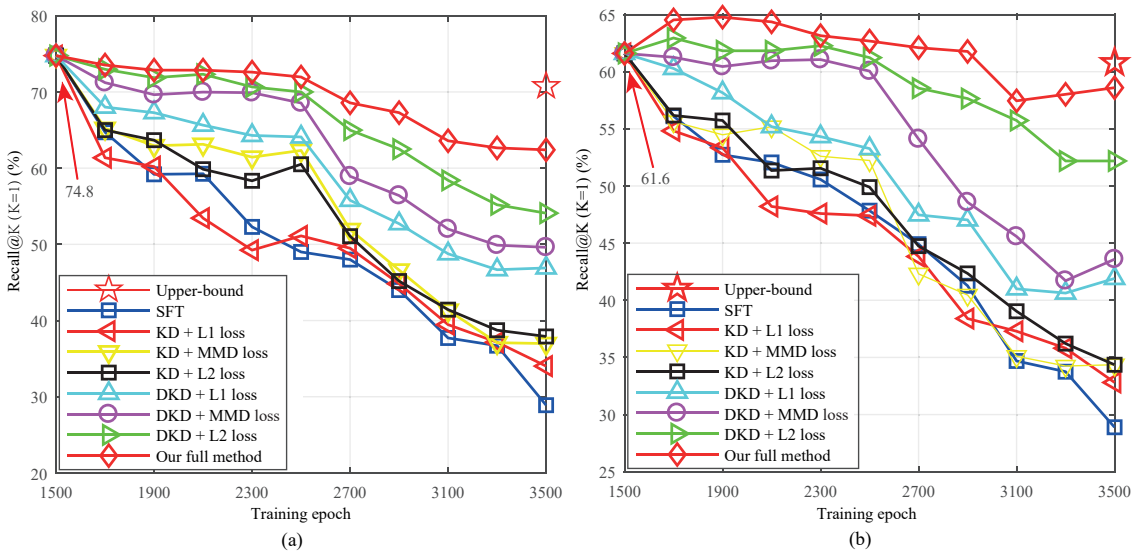


Figure 7.3: The performance degradation evaluation on the CUB-200 dataset: (a) on the seen set; and (b) on the unseen set.

Stanford-Dogs), which are shown in Table 7.4. The "Student model" refers to the model trained by our DKD. We observe that this on-the-fly teacher achieves good generalization performance on the new task.

(4) Evaluation of the generated images. One benefit of using BatchNorm layers is that the representative images can be directly generated using the frozen teacher, without any other operations or additional generative networks. For evaluation, we select the generated images by using the frozen teacher trained on CUB-200, evaluated by using the inception score [262] and Fréchet Inception Distance (FID) [263]. The origin images are chosen randomly from previous 70 classes (4076 images) on CUB-200. These class labels are used to generate equal representative images. As shown in Table 7.5, these results demonstrate that the efficacy of loss terms L_{BN} and $L_{cluster}$ for generating images. Moreover, several generated birds images for the CUB-200 dataset are visualized in Figure 7.4. The generated representative images for the Stanford-Dogs dataset are listed in Figure 7.5). As required by lifelong image retrieval, this student needs to remember previously learned knowledge and capture new information on the new dataset (*i.e.* Stanford-Dogs). As a result, the images generated by this trained student model share some properties for Birds images and Dog images. Similarly, the representative images generated for the Stanford-Cars dataset are shown in Figure 7.6. We observe that these representative images show more semantics for the Cars images. The reason is that the student is prone to learning new information on the Stanford-Cars dataset. Furthermore, the image generation process on the CUB-200 dataset is illustrated in Figure 7.7. The initial input is random Gaussian noise, which is optimized iteratively until $Iter = 2000$, as can be seen in Algorithm 2.

(5) Ablation study. We perform an ablation analysis of the proposed method,

Table 7.3: Recall@K (K=1) comparison (%) of s and u on three datasets. The results are reported when the model is trained on Stanford-Cars and then tested backward on the previous two datasets. ‡ refers to BatchNorm statistics are used for enhancing the knowledge distillation using the frozen teacher only. Likewise, ‡ refers to BatchNorm statistics are used to enhance the frozen teacher. The best balanced results are highlighted in boldface.

	CUB-200 → Stanford-Dogs → Stanford-Cars								
	Test on CUB-200			Test on Stanford-Dogs			Test on Stanford-Cars		
	s	u	H	s	u	H	s	u	H
Recall@K	K=1	K=1	K=1	K=1	K=1	K=1	K=1	K=1	K=1
SFT [226]	28.9	28.1	28.5	40.6	63.3	49.5	72.6	78.1	75.3
KD+L_1 loss [216]	34.0	32.8	33.4	44.5	68.3	53.9	71.8	79.3	75.4
KD+L_{mmd} loss[37]	37.0	34.4	35.7	46.1	69.7	55.5	72.0	76.9	74.4
KD+L_2 loss[217]	37.9	34.4	36.1	43.8	67.8	53.2	74.9	80.8	77.7
KD+ KL div.	37.3	34.3	35.7	45.9	69.1	55.2	71.9	80.6	76.0
KD+L_1 loss†	69.7	58.5	63.6	44.2	74.2	55.4	37.9	58.1	45.9
KD+L_{mmd} loss†	70.7	60.8	65.4	47.9	76.1	58.8	40.3	58.4	47.7
KD+L_2 loss†	70.9	62.3	66.3	53.8	79.8	64.3	40.2	58.3	47.6
KD+KL div.†	71.1	65.6	68.2	55.8	80.2	65.8	40.8	58.9	48.2
DKD+L_1 loss‡	46.9	41.9	44.3	59.5	77.8	67.4	74.1	80.8	77.3
DKD+L_{mmd}‡	49.6	43.6	46.4	58.7	77.4	66.8	71.5	78.9	75.0
DKD+L_2 loss‡	54.1	52.2	53.1	58.8	78.6	67.3	**75.1**	80.8	77.9
DKD+KL div.‡	**62.4**	**58.6**	**60.5**	**67.4**	**84.3**	**74.9**	73.2	**83.7**	**78.1**
Joint training	71.5	62.5	66.7	71.2	83.3	76.8	74.3	81.6	77.8

Table 7.4: Evaluation for the on-the-fly teacher on the second task.

	CUB-200 → Stanford-Dogs		
	Test on Stanford-Dogs		
	s	u	H
Recall@K	K=1	K=1	K=1
Fine-tuning	72.2	84.9	78.0
Student model	74.4	86.5	80.0
On-the-fly teacher	74.6	86.3	80.0
Joint training	74.5	86.7	80.1

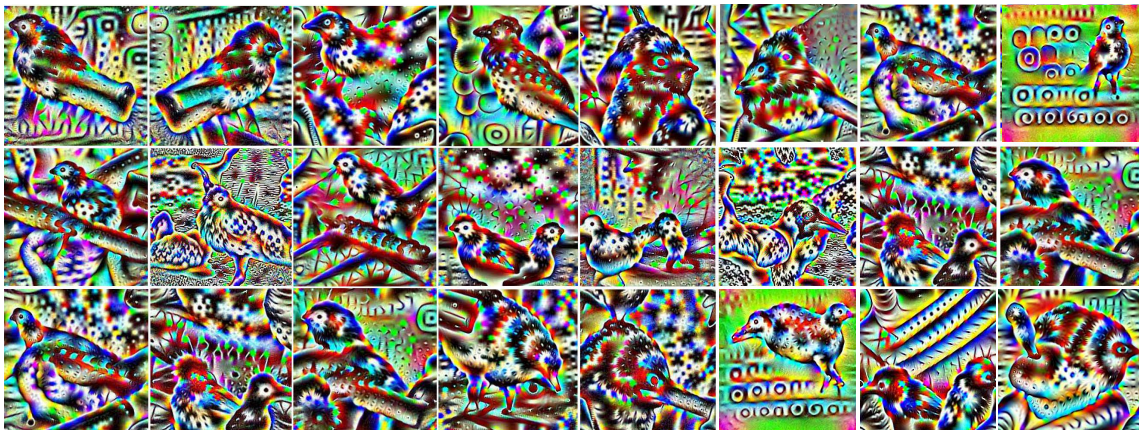


Figure 7.4: The generated representative images for CUB-200.

Table 7.5: Evaluation of the generated images

	Inception score	FID
Input random noise	0.93±0.01	401
Generated birds images	3.09±0.39	198
Origin birds images	5.24±0.30	0

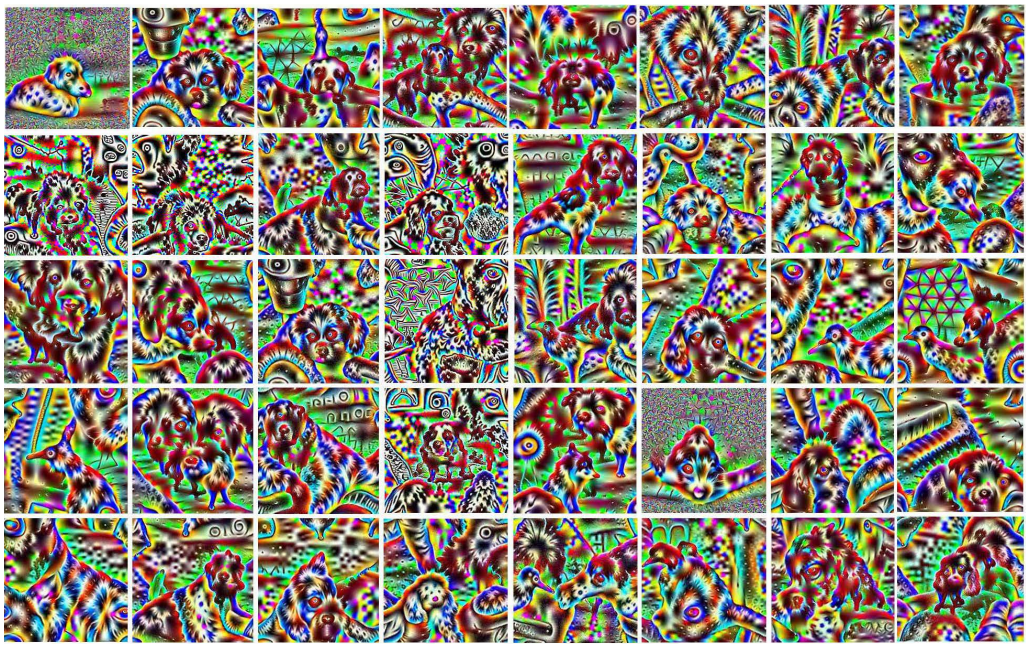


Figure 7.5: The generated images for the Stanford-Dogs dataset.

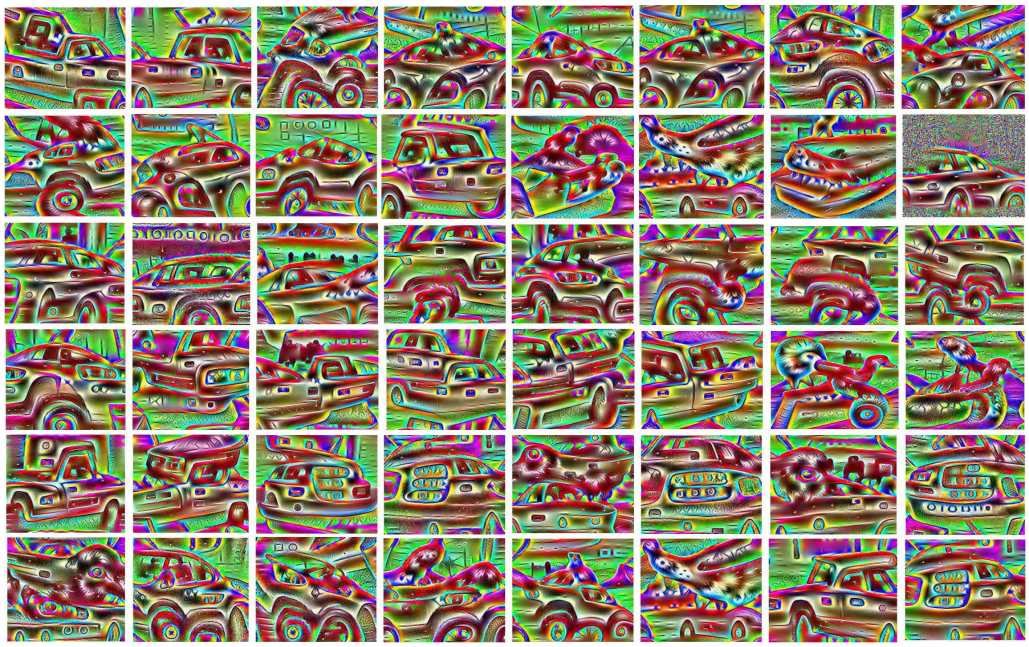


Figure 7.6: The generated images for the Stanford-Cars dataset.

as shown in Table 7.6. Consistent to previous experiments, we use the sequence of two tasks: CUB-200 → Stanford-Dogs. We build the fine-tuning method as a

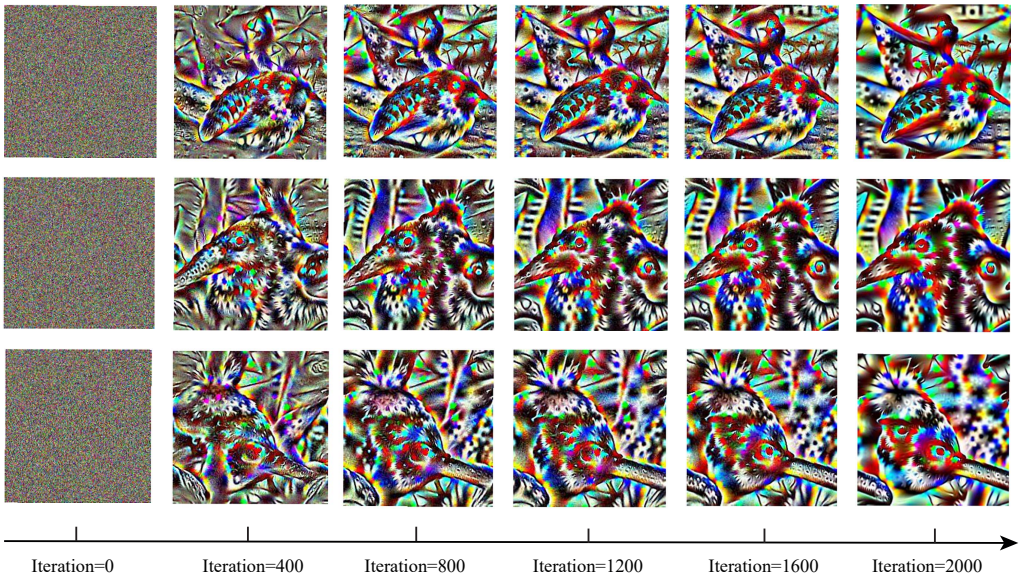


Figure 7.7: Illustration of image generation process on the CUB-200 dataset.

Baseline by using L_s only. As noted, the baseline model suffers from forgetting on the first task. **Case 1** is the knowledge distillation using L_{kd_1} from the frozen teacher only. As a result, the previously learned knowledge is transferred to the student (improving R@K=1 from 51.4% to 56.7% on CUB-200). To demonstrate the efficacy of BatchNorm statistics, we study **Case 2** where representative images are generated using $(L_{BN} + L_{cluster})$. Compared to *Case 1*, the student trained under this condition is prone to the first task and has its performance improved from 56.7% to 68.3% significantly, while performance on the second task degrade from 78.9% to 68.7%. **Case 3** is designed for the scenario where the self-motivated student is regularized only by the on-the-fly teacher when learns the second task. Consequently, the student improves on the second task (from 78.0% to 79.6%) and keeps the performance on the first task similar to the Baseline. We explore **Case 4** to study the importance of self-motivated learning of the student, which is regularized by dual knowledge distillation, but without using L_s. As a result, the student remembers the previous knowledge well and has a good generalization accuracy Recall@1 of 76.6% on the second task. Furthermore, **Case 5** refers to the network is regularized by two teachers but without using the BatchNorm statistics to enhance the frozen teacher. Compared to *Case 3*, the student improves its performance on the first task (*e.g.* from 50.8% to 56.9%), while the performance on the second task is kept unchanged. Finally, when the student is self-motivated to learn by using the term L_s, *i.e.* our DKD full method, whose generalization performance is improved from 76.6% in *Case 4* to 80.0% while the performance on the first task is close to the reference.

Table 7.6: Ablation study for lifelong image retrieval on the two-task setup. As defined in Eqs. 7.4 and 7.5, the representative image generation process is constrained by $L_g = L_{BN} + L_{cluster}$.

		CUB-200 → Stanford-Dogs					
		Test on CUB-200			Test on Stanford-Dogs		
		s	u	H	s	u	H
	Recall@K	K=1	K=1	K=1	K=1	K=1	K=1
Baseline	Fine-tuning by using L_s	56.0	47.5	51.4	72.2	84.9	78.0
Case 1	$L_s + L_{kd_1}$	62.2	52.1	56.7	73.6	85.0	78.9
Case 2	$L_s + L_{kd_1} + L_g$	73.5	63.8	68.3	60.0	80.3	68.7
Case 3	$L_s + L_{kd_2} + L_{te_2}$	55.1	47.1	50.8	74.0	86.2	79.6
Case 4	$L_{kd_1} + L_g + L_{kd_2} + L_{te_2}$	73.2	62.4	67.3	69.0	86.1	76.6
Case 5	$L_s + L_{kd_1} + L_{kd_2} + L_{te_2}$	59.7	54.5	56.9	74.0	86.2	79.6
Ours	$L_s + L_{kd_1} + L_g + L_{kd_2} + L_{te_2}$	72.0	62.7	67.0	74.4	86.5	80.0
Reference	Joint training by using L_s	74.1	64.1	68.7	74.5	86.7	80.1

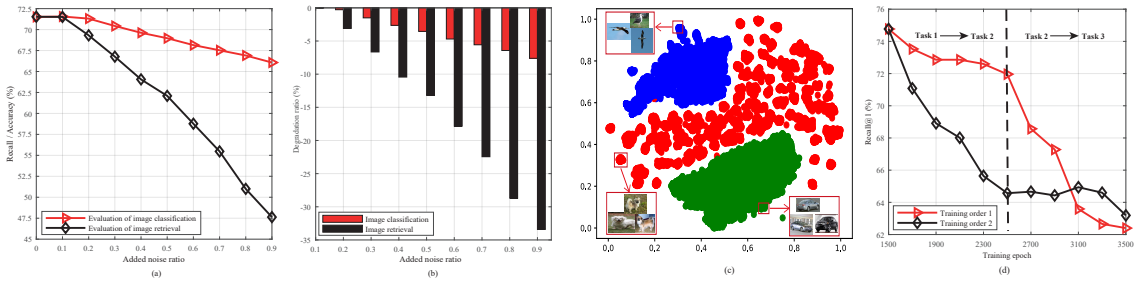


Figure 7.8: Sensitivity comparisons of image classification and image retrieval. (a) Recall rate / classification accuracy; (b) Performance degradation ratios for different noise ratios. (c) Dataset distributions visualization; (d) Performance evolution of two training orders, evaluated on the first task, *i.e.* on the CUB-200 dataset.

7.5.4 Further explorations

(1) Comparison with classification-based tasks. In terms of reducing forgetting, we observe that lifelong image retrieval is more challenging than classification-based tasks that focus on classification probabilities. The classification model is more stable, as long as image features of old data are classified within the range of prior boundaries, whereas image retrieval is sensitive to the matching between features. A small change in features would have a significant impact on feature matching. This makes the problem of minimizing forgetting more difficult. As a demonstration, we build an additional classifier on top of the fully-connected layers and use the LwF method [212] to train under the sequence: CUB-200 → Stanford-Dogs. During testing, we sample Gaussian noise from $\mathcal{N}(0, 0.1)$ and add it to each image, which affects the retrieval features and the final classification probabilities of the same model. We vary the ratio of the Gaussian noise and consider the evolution of retrieval recall and classification accuracy on the seen set of CUB-200. The results are reported in Figure 7.8. As can be seen, image retrieval task is more sensitive than image classification task for same levels of noise distraction.

(2) Training order exploration. We consider the training order 1: CUB-200 → Stanford-Dogs → Stanford-Cars in Table 7.3. To examine the effect of the task training order, we keep starting with CUB-200 and explore the other training order 2: CUB-200 → Stanford-Cars → Stanford-Dogs. We visualize all training samples of three datasets in Figure 7.8(c). For the two training orders, we evaluate the performance on the seen set of the first task (*i.e.* CUB-200) by using the model trained at the end of tasks (*i.e.* Stanford-Cars and Stanford-Dogs). The results are depicted in Figure 7.8(d). In general, the model suffers from performance degradation with respect to these two training orders. Due to the different distributions of datasets, the training order affects the performance greatly. In case of training order 1, the samples from Stanford-Dogs on task 2 are distributed closely to the samples from CUB-200. Therefore, the degradation during the "task 1 → task 2" session is relatively slow. However, the vehicle images from task 3 are distributed farther away from the bird images in task 1, which causes serious degradation during the "task 2 → task 3" session. In contrast, for training order 2, the performance degrades significantly from CUB-200 to Stanford-Cars during the "task 1 → task 2" session and whereas it becomes slow again during the "task 2 → task 3" session.

7.6 Chapter Conclusions

In this chapter, we explored image retrieval in a lifelong scenario and considered reducing catastrophic forgetting and simultaneously improving generalization performance. This goal is achieved by training a dual knowledge distillation framework to transfer previously learned knowledge and newly captured information. We used the stored statistics in the BatchNorm layers of the frozen teacher to generate representatives images to further reduce catastrophic forgetting on preceding tasks. The efficacy of the proposed method was demonstrated by thorough experimental results on three datasets. A limitation of this work is that the semantic drifts between training data in the task sequence still result in significant forgetting. In future work, more efficient approaches need to be investigated to realize lifelong image retrieval without forgetting. Furthermore, it would be very valuable to explore lifelong image retrieval on non-fine grained datasets or practical applications such as commercial shopping and recommendation systems.

Chapter 8

New Ideas and Trends in Deep Multimodal Content Understanding

In previous chapters, we focused the research on image retrieval and cross-modal retrieval in the context of non-incremental or incremental learning. In the past years, deep learning has also been explored for the field of multimodal learning.

In this Chapter, we present the recent new ideas and trends in multimodal content understanding filed, focusing on the analysis of two modalities: image and text. These new methods can be further used for intelligent image retrieval to seek performance improvement. Unlike classic reviews of deep learning where unimodal image classifiers such as VGG, ResNet, and Inception module are central topics, this chapter examines recent multimodal deep models and structures, including autoencoders, generative adversarial nets and their variants. These models go beyond the simple image classifiers in which they can do uni-directional (*e.g.* image captioning, image generation) and bi-directional (*e.g.* cross-modal retrieval, visual question answering) multimodal tasks. Besides, we analyze two aspects of the challenge in terms of better content understanding in deep multimodal applications. We then introduce current ideas and trends in deep multimodal feature learning, such as feature embedding approaches and objective function design, which are crucial in overcoming the aforementioned challenges.

Keywords

Multimodal deep learning, Ideas and trends, Content understanding

This chapter is based on the following publication [39]:

- Chen, W., Wang, W., Liu, L., and Lew, M.S., "New Ideas and Trends in Deep Multimodal Content Understanding: A Review." Neurocomputing, 2020, pp. 195-215.

8.1 Introduction

Multimodal content understanding aims at recognizing and localizing objects, determining the attributes of objects, characterizing the relationships between objects, and finally, describing the common semantic content among different modalities. In the information era, rapidly developing technology makes it more convenient than ever to access a sea of multimedia data such as text, image, video, and audio. As a result, exploring semantic correlation to understand content for diverse multimedia data has been attracting much attention as a long-standing research field in the computer vision community.

Recently, the topics range from speech-video to image-text applications. Considering the wide array of topics, we restrict the scope of this survey to image and text data specifically in the multimodal research community, including tasks at the intersection of image and text (also called cross-modal). According to the available modality during testing stage, multimodal applications include bi-directional tasks (*e.g.* image-sentence search [264], visual question answering (VQA) [265]) and uni-directional tasks (*e.g.* image captioning [266], image generation [22, 267]), both of them will be introduced in the following sections.

Data from visual and textual modalities are represented as unimodal features using domain-specific networks. Complementary information from these unimodal features is appealing for multimodal content understanding. For example, the unimodal features can be further projected into a common space by using another neural network for an vision task. For clarity, we illustrate the flowchart of deep multimodal research in Figure 8.1. On the one hand, the neural networks are comprised by successive linear layers and non-linear activation functions, the image or text data is represented in a high abstraction way, which is helpful for reducing the "semantic gap" [10], as defined in Chapter 3. On the other hand, different modalities are characterized by different statistical properties. Image is 3-channel RGB array while text is often symbolic. When represented by different neural networks, their features have unique distributions and differences, which leads to the "heterogeneity gap" [176]. To understand multimodal content, deep neural networks should be able to reduce the difference between high-level semantic concepts and low-level features in intra-modality representations, as well as construct a common latent space to associate semantic correlations in inter-modality representations.

Much effort has gone into mitigating these two challenges to improve content understanding. Some works involve deep multimodal structures such as cycle-consistent reconstruction [268, 269], while others focus on feature extraction nets such as graph convolutional networks [270, 271]. In some algorithms, reinforcement learning is combined with deep multimodal feature learning [272, 273]. These recent ideas are the scope of this chapter.

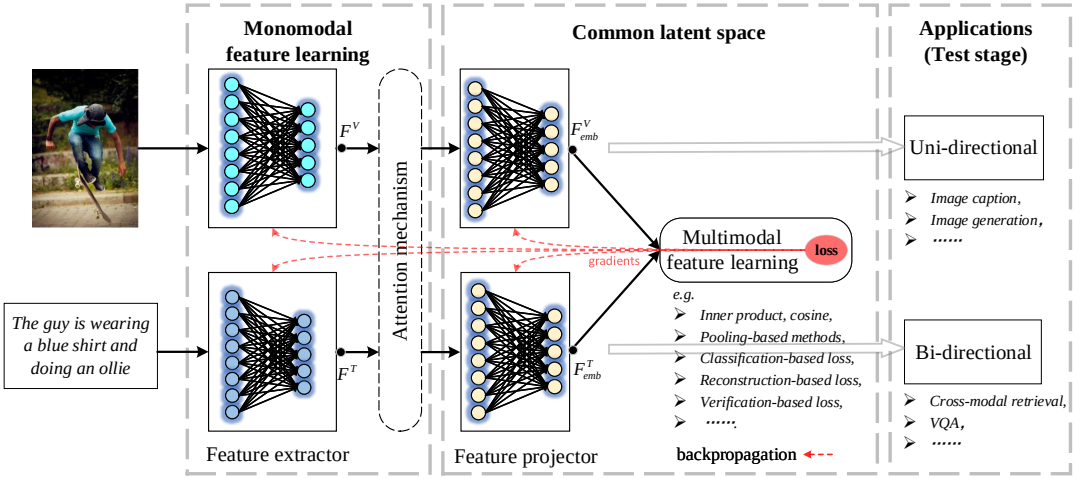


Figure 8.1: A general flowchart of deep multimodal feature learning.

8.2 Multimodal Applications

This section aims to summarize various multimodal applications where image and text data are involved. These applications have gained a lot of attention lately and show a natural division into uni-directional and bi-directional groups. The difference is that for uni-directional scenarios only one modality is available at the test stage, whereas in bi-directional scenarios, two modalities are required.

8.2.1 Uni-directional applications

a. Image-to-text tasks

Image captioning is a task that generates a sentence description for an image and requires recognizing important objects and their attributes, then inferring their correlations within the image [274]. After capturing these correlations, the captioner yields a syntactically correct and semantically relevant sentence. To understand the visual content, images are fed into convolutional neural networks to learn hierarchical features, which constitutes the feature encoding process. The produced hierarchical features are transformed into sequential models (*e.g.* RNN, LSTM) to generate the corresponding descriptions. Subsequently, the evaluation module produces description difference as the feedback signals to update the performance of each block. Deep neural networks are commonly used in image captioning. In the following sections, we will examine the methods widely used to improve image captioning performance, including evolutionary algorithm [275], generative adversarial networks [180, 276], reinforcement learning [272, 273], memory networks [277, 278], and attention mechanisms [279, 280].

According to captioning principles, researchers focus on specific caption generation tasks, such as image tagging [281], visual region captioning [282], and object captioning [283]. Analogously, these tasks are also highly dependent on the regional

image patch and sentences/phrases organization. The specific correlations between the features of objects (or regions) in one image and the word-level (or phrase-level) embeddings are explored instead of global dependence of the holistic visual and textual features.

b. Text-to-image tasks

Compared to generating a sentence for a given image, generating a realistic and plausible image from a sentence is even more challenging. Namely, it is difficult to capture semantic cues from a highly abstract text, especially when the text is used to describe complex scenarios as found in the MS-COCO dataset [192, 284]. Text-to-image generation is such a kind of task which maps from textual modality to visual modality.

Text-to-image generation requires synthesized images to be photo-realistic and semantically consistent (*i.e.* preserving specific object sketches and semantic textures described in text data) through architectures such as Variational Auto-Encoders (VAE) [285], auto-regressive models [286] and Generative Adversarial Networks (GANs) [22, 180]. One example is to generate a semantic layout as intermediate information from text data to bridge the heterogeneity gap in image and text [287, 288]. Some works focus on the network structure design for feature learning. For image synthesis, novel derivative architectures from GANs [180] have been explored in hierarchically nested adversarial networks [289], perceptual pyramid adversarial networks [290], iterative stacked networks [23, 291], attentional generative networks [292, 293], cycle-consistent adversarial networks [268], and symmetrical distillation networks [294].

One of limitations of image generation is that, while generation models work well and achieve promising results on single category object datasets like Caltech-UCSD CUB [295] and Oxford-102 Flower [295], existing methods are still far from promising on complex dataset like MS-COCO where one image contains more objects and is described by a complex sentence. To compensate for this limitation, word-level attention [292], hierarchical text-to-image mapping [288] and memory networks [296] have been explored. In the future, one direction may be to make use of the Capsule idea proposed by Hinton [297] since capsules are designed to capture the concepts of objects.

8.2.2 Bi-directional applications

a. Cross-modal retrieval

Cross-modal retrieval has been researched for decades. The aim is to return the most relevant image (text) when given a query text (image). There are two aspects should be considered: retrieval accuracy and retrieval efficiency.

For the first, it is desirable to explore semantic correlations across an image and text features. To meet this requirement, the aforementioned heterogeneity gap and the semantic gap are the challenges to deal with. Some novel techniques that have been proposed are as follows: attention mechanisms and memory networks are employed to align relevant features between image and text [298]; Bi-directional sequential models (*e.g.* Bi-LSTM [188]) are used to explore spatial-semantic correlations [264]; Graph-based embedding and graph regularization are utilized to keep semantic order in text feature extraction process [299]; Information theory is applied to reduce the heterogeneity gap in cross-modal hashing [34]; Adversarial learning strategies and GANs are used to estimate common feature distributions [177, 300].

For the second, recent hashing methods have been explored owing to the computation and storage advantages of binary code [178]. Essentially, methods such as attention mechanisms and adversarial learning [178] are applied for learning compact hash codes with different lengths. However, the problems should be considered when one employs hashing methods for cross-modal retrieval are feature quantization and non-differential binary code optimization. Focusing on the feature quantization, Wang *et al.* [301] introduce a hashing code learning algorithm in which the binary codes are generated without relaxation so that the large quantization and non-differential problems are avoided.

There still exists room for performance improvement (see Figure 8.4-8.5). For example, to employ graph-based methods to construct semantic information within two modalities, more context information such as objects link relationships are adopted for more effective semantic graph construction.

b. Visual question answering

Visual question answering (VQA) is a challenging task in which an image and a question are given, then a correct answer is inferred according to visual content and syntactic principle. Since VQA was proposed, it has received increasing attention in recent years. For example, there are some training datasets [302] built for this task, and some network training tips and tricks are presented in work [303].

To infer correct answers, VQA systems need to understand the semantics and intent of the questions completely, and also should be able to locate and link the relevant image regions with the linguistic information in the questions. VQA applications present two-fold difficulties: feature fusion and reasoning rationality. Thus, VQA more closely reflects the difficulty of multimodal content understanding, which makes VQA applications more difficult than other multimodal applications. Compared to other applications, VQA has various and unknown questions as inputs. Specific details (*e.g.* activity of a person) in the image should be identified along with the undetermined questions. Moreover, the rationality of question answering is based on high-level knowledge and advanced reasoning capability of deep models.

The research on VQA includes: free-form open-ended questions [304], where the answer could be words, phrases, and even complete sentences; object counting questions [305] where the answer is counting the number of objects in one image; multi-choice questions [279] and Yes/No binary problems [306]. In principle, the type of multi-choice and Yes/No can be viewed as classification problems, where deep models infer the candidate with maximum probability as the correct answer. These two types are associated with different answer vocabularies and are solved by training a multi-class classifier. In contrast, object counting and free-form open-ended questions can be viewed as generation problems [302] because the answers are not fixed, only the ones related to visual content and question details.

8.3 Recent Advances in Content Understanding

Lots of remarkable progresses about exploring content understanding between image and text have been made. In general, these advances are mainly from a viewpoint of network structure and a viewpoint of feature extraction/enhancement. To this end, combining the natural process pipeline of multimodal research (see Figure 8.1), we categorize these research ideas into three groups: deep multimodal structures presented in Section 8.3.1, multimodal feature extraction approaches introduced in Section 8.3.2, and common latent space learning described in Section 8.3.3.

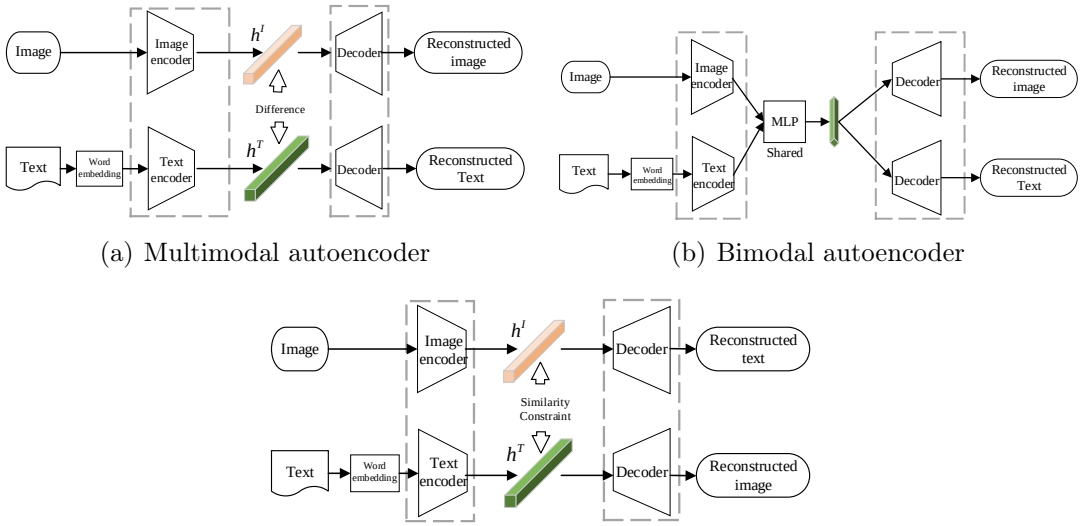
8.3.1 Deep multimodal structures

Deep multimodal structures are the fundamental frameworks to support different deep networks for exploring visual-textual semantics. These frameworks have critical influences for the following feature extraction and common latent space learning. To understand the semantics between images and text, deep multimodal structures usually involve computer vision and natural language processing (NLP) field [307]. During the past years, a variety of related methods have blossomed and accelerated the performance of multimodal learning directly in multimodal applications.

Deep multimodal structures include generative models, discriminative models. Generative models implicitly or explicitly represent data distributions measured by a joint probability $P(X, Y)$, where both raw data X and ground-truth labels Y are available in supervised scenarios. Discriminative models learn classification boundaries between two different distributions indicated by conditional probability $P(Y|X)$. Recent representative network structures for multimodal feature learning are auto-encoders and generative adversarial networks.

a. AutoEncoders

The main idea of auto-encoder is to first encode data from a source modality as hidden representations and then to use a decoder to generate features (or data) for the target modality. Thus, it is commonly used for bi-directional applications where



(a) Multimodal autoencoder (b) Bimodal autoencoder

(c) Correlation autoencoder

Figure 8.2: Convolutional autoencoder used for deep multimodal learning. The branch for image feature learning can adopt hierarchical networks such as CNNs; the branch for text feature learning can capture the dependency relations in a sentence by sequential models such as RNN and LSTM. Usually, a reconstruction loss function is used to optimize network training.

two modalities are available at the test stage. For this structure, reconstruction loss is the constraint for training encoder and decoder to well capture the semantic correlations between image and text features. For clarity, we identify three ways for correlation learning using auto-encoders in Figure 8.2. For instance, as shown in Figure 8.2(a), the input images and text are processed separately with non-shared encoder and decoder, after which these hidden representations from the encoder are coordinated through a constraint such as Euclidean distance [308]. The coordinated methods can be replaced by joint methods in Figure 8.2(b) where image and text features are projected into a common space with a shared multilayer perceptron (MLP). Subsequently, the joint representation is reconstructed back to the original raw data [309]. Alternatively, feature correlations are captured by cross reconstruction with similarity constraints between hidden features. The idea of constraining sample similarity is also incorporated with GANs into a cycle-consistent formation for cross-modal retrieval such as CYC-DGH [269].

The neural networks contain in the encoder-decoder framework can be modality specific. For image data, the commonly used neural networks are CNN while sequential networks like LSTM are most often used for text data. When applied for multimodal learning, the decoder (*e.g.* LSTM) constructs hidden representations of one modality in another modality. The goal is not to reduce reconstruction error but to minimize the output likelihood estimation. Therefore, most works focus on the decoding since it is a process to project the less meaningful vectorial representations to meaningful outputs in target modality. Under this idea, several extensions have

been introduced. The main difference among these algorithms lies in the structure of the decoder. For example, "stack and parallel LSTM" [310] is to parallelize more parameters of LSTMs to capture more context information. Similar ideas can be found in "CNN ensemble learning" [311]. Instead of grabbing more information by stacking and paralleling, "Attention-LSTM" [310] combines attention technique into LSTM to highlight most relevant correlations. An adversarial training strategy is employed into the decoder to make all the representations discriminative for semantics but indiscriminative for modalities so that intra-modal semantic consistency is effectively enhanced [309]. Considering the fixed structure in the decoder like RNN might limit the performance, Wang *et al.* [275] introduce evolutionary algorithm to adaptively generate neural network structures in the decoder.

b. Generative adversarial networks

Adversarial learning from generative adversarial networks [180] has been employed into applications including image captioning [312], cross-modal retrieval [309] and image generation [23, 289, 291], but has been less popular in VQA tasks. GANs combine generative sub-models and discriminative sub-models into a unified framework in which two components are trained in an adversary manner.

GANs can cope with the scenarios where there are some missing data. To accurately explore the correlations between two modalities, multimodal research works involving GANs have been focusing on the whole network structure and its two components: *generator* and *discriminator*.

For the generator which also can be viewed as an encoder, an attention mechanism is often used to capture the important key points and align cross-modal features such as Attention-aware methods [292]. Sometimes, Gaussian noise is concatenated with the generator's input vector to improve the diversity of generated samples and avoid model collapse, such as the conditioning augmentation block in StackGAN [23]. To improve its capacity for learning hierarchical features, a generator can be organized into different nested structures to capture multi-level semantics such as hierarchical-nested [289] and hierarchical-pyramid [290].

The discriminator, which usually performs binary classification, attempts to discriminate the ground-truth labels from the outputs of the generator. Some recent ideas are proposed to improve the discrimination of GANs. Originally, discriminator in the first work [180] just needs to classify different distributions into "*True*" or "*False*" [22]. However, discriminator can also make a class label classification where a label classifier is added on the top of discriminator [313]. Apart from the label classification, a semantic classifier is designed to further predict semantic relevances between a synthesized image and a ground-truth image for text-to-image generation [267]. Only focusing on the paired samples leads to relatively-weak robustness. Therefore, the unmatched image-text samples can be fed into a discriminator (*e.g.*

GAN-INT-CLS [22] and AACR [300]) so that the discriminator would have a more powerful discriminative capability.

The application of GANs in multimodal research are categorized into direct methods [22, 313], hierarchical methods [289, 290], and iterative methods [23, 291, 292]. Contrary to direct methods, hierarchical methods divide raw data in one modality (*e.g.* image) into different parts such as a "style" and "structure" stage, thereby, each part is learned separately. Alternatively, iterative methods separate the training into a "coarse-to-fine" process where details of the results from a previous generator are refined. Besides, cycle-consistency [314] is introduced for unsupervised image translation where a self-consistency (reconstruction) loss tries to retain the patterns of input data after a cycle of feature transformation. This idea is then applied into tasks like image generation [268] and cross-modal retrieval [269] to learn semantic correlation in an unsupervised way.

In recent years, adversarial learning is widely used to design algorithms for deep multimodal learning [177, 178, 309]. For these algorithms, there are no classifiers for binary classification. Instead, two sub-networks are trained with the constraints of competitive loss functions. As the dominant popularity of adversarial learning, some works are performed by combining auto-encoders and GANs in which the encoder in auto-encoders and the generator in GANs share the same sub-network [309, 315, 316]. For example, in the first work about unsupervised image captioning [316], the core idea of GANs is used to generate meaningful text features from scratch of text corpus and cross-reconstruction is performed between synthesized text features and true image features.

8.3.2 Multimodal feature extraction

Feature extraction is closer for exploring visual-textual content relations, which is the prerequisite to discriminate the complementarity and redundancy of multiple modalities. In this section, we introduce several effective multimodal feature extraction methods for addressing the heterogeneity gap. In general, these methods focus on (1) learning the structural dependency information to reasoning capability of deep neural networks and (2) storing more information for semantic correlation learning during model execution. Moreover, (3) feature alignment schemes using attention mechanism are also widely explored for preserving semantic correlations.

a. Graph embeddings with graph convolutional networks

Words in a sentence or objects within an image have some dependency relationships, and graph-based visual relationship modelling is beneficial for the characteristic. Graph Convolutional Networks (GCNs) are alternative neural networks designed to capture this dependency information. Compared to standard neural networks such as CNNs and RNNs, GCNs would build a graph structure which models a set of objects (nodes) and their dependency relationships (edges) in an image or

sentence, embed this graph into a vectorial representation, which is subsequently integrated seamlessly into the follow-up steps for processing. Graph representations reflect the complexity of sentence structure and are applied to natural language processing such as text classification [154]. For deep multimodal learning, GCNs receive increasing attention and have achieved breakthrough performance on several applications, including cross-modal retrieval [317], image captioning [270, 271, 318], and VQA [319].

Graph convolutional networks in multimodal learning can be employed in text feature extraction [317, 319] and image feature extraction [270, 271, 318]. Among these methods, GCNs capture semantic relevances of intra-modality according to the neighborhood structure. GCNs also capture correlations between two modalities according to supervisory information. Note that vector representations from graph convolutional networks are fed into subsequent networks (*e.g.* "encoder-decoder" framework) for further learning.

GCNs are introduced to determine the attributes and subsequently characterize the relationships between image and text [319]. To use GCNs, an image is parsed into different objects, scenes, and actions. Also, a corresponding question is parsed and processed to obtain its question embeddings and entity embeddings. These embedded vectors of image and question are concatenated into node embeddings then fed into graph convolutional networks for semantic correlation learning. Finally, the output activations from GCNs are fed into sequential networks to predict answers.

As an alternative method, GCNs are worthy more exploration for correlations between two modalities. Moreover, there exist two limitations in GCNs. On the one hand, graph construction process is overall time- and space-consuming; On the other hand, the accuracy of output activations from GCNs mostly relies on supervisory information to construct an adjacency matrix by training, which are more suitable for structured data, so flexible graph embeddings for image and/or text remains an open problem.

b. Memory-augmented networks

To enable deep networks to understand multimodal content and have better reasoning capability for various tasks, another solution that has gained attention recently is memory-augmented networks. Directly, when much information in mini-batch even the whole dataset is stored in a memory bank, such networks have greater capacity to memorize correlations.

In conventional neural networks like RNNs for sequential data learning, the dependency relations between samples are captured by the internal memory of recurrent operations. These recurrent operations might be inefficient in understanding and reasoning overextended contexts or complex images. For instance, most captioning models are equipped with RNN-based encoders, which predict a word at every time

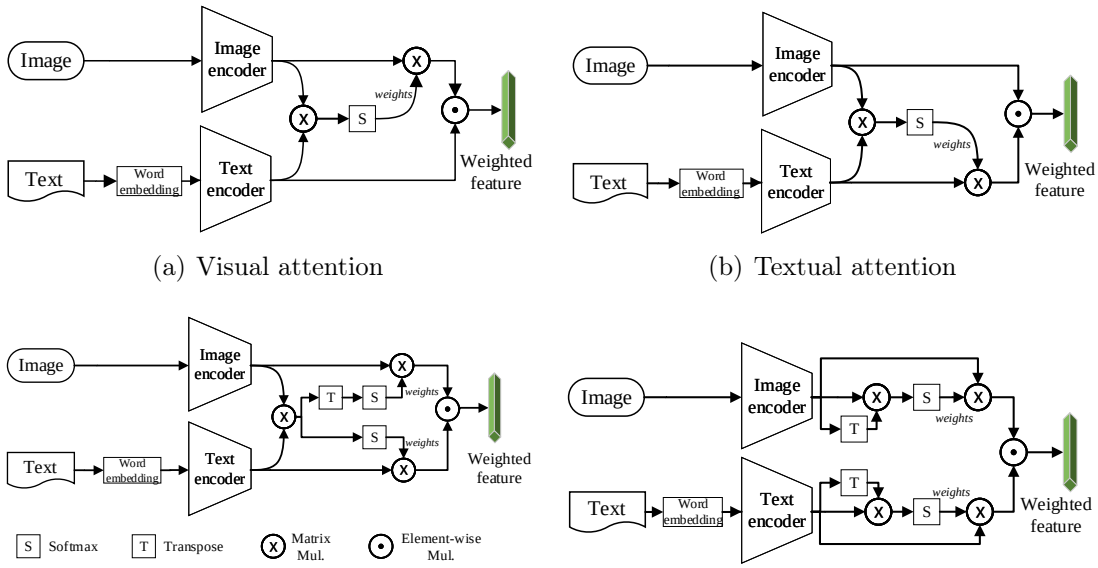
step based only on the current input and hidden states used as implicit summaries of previous histories. However, RNNs and their variants often fail to capture long-term dependencies [278]. For this limitation, memory networks [277] are introduced to augment the memory primarily used for text question-answering [320]. Memory networks improve understanding of both image and text, and then "remember" temporally distant information.

Memory-augmented networks are used in cross-modal retrieval [298], image captioning [278], and VQA [321]. Memory-augmented networks can be regarded as recurrent neural networks with explicit attention methods that select certain parts of the information to store in their memory slots. The memory slots are a kind of external memory to support learning. During training, a network such as LSTM or GRU, which acts as a memory controller, refers to these memory slots to compute reading weights. According to the weights, the essential information is obtained to predict the output sequence. Meanwhile, the controller computes writing weights to update values in memory slots for the next time-step of the training [322].

The performance of memory networks relates to the memory slots' initialization strategy and the stored information. For this aspect, memory networks have been combined with other techniques like attention mechanisms [323] to further improve its feature learning capability. For example, Xiong *et al.* [324] explore the impact of different initialization strategies to demonstrate that initializations from the outputs of pre-trained networks have better performance. This was verified in works [325] where output features from image patches are stored into memory slots of spatial memory networks for VQA. Thereby, generated answers are updated based on gathering evidence from the accessed regions in memory slots. Similarly, Ma *et al.* [326] adopt LSTM to obtain text features of each sentence and store into memory slots. Then memory-augmented networks are utilized to determine the importance of concatenated visual and text features over the whole training data. Further considering both two modalities, a visual knowledge memory networks is introduced in which memory slots store key-value vectors computed from images, query questions and a knowledge base [321]; Instead of storing the actual output features, Song *et al.* [298] adopt memory slots to store a prototype concept representation from pre-trained concept classifiers, which is inspired from the process of human memory.

c. Attention mechanism for deep multimodal learning

Attention mechanisms are widely used to tackle this issue in various multimodal tasks, such as VQA [327, 328, 329] and image captioning [266, 270, 280]. In principle, the attention mechanisms compute different importances according to relevances between two global (or local) multimodal features and assign different importances to these features. Thereby, the networks are more targeted at the sub-components of the source modality–regions of an image or words of a sentence. To further explore the relevances between two modalities, the attention mechanisms are adopted



(a) Visual attention

(b) Textual attention

(c) Co-attention

(d) Self-attention

Figure 8.3: Diagram for different types of attention mechanisms used in deep multimodal learning.

on multi-level feature vectors [325], employed in a hierarchical scheme [330], and incorporated with graph networks for modelling semantic relationships.

To elaborate on the current ideas and trends of attention algorithms, we categorize this popular mechanism into different types. According to objective computing vectors, we categorize the current attention algorithms into four types: visual attention, textual attention, co-attention, and self-attention, as illustrated in Figure 8.3. We further categorize the attention algorithms into single-hop and multiple-hop (*i.e.* stacked attention) according to the iterations of importance calculation.

Visual attention. As shown in Figure 8.3(a), visual attention schemes are used in scenarios where text features (*e.g.* from a query question) are used as context to compute their co-relevance with image features, and then the relationships are used to construct a normalized weight matrix. Subsequently, this matrix is applied to original image features to derive text-guided image features using element-wise multiplication operation (linear operation). The weighted image features have been aligned by the correlation information between image and text. Finally, these aligned multimodal features are utilized for prediction or classification. This idea is common in multimodal feature learning [266, 270, 279, 280, 325, 327, 328] and has been incorporated to get different text-guided features. For example, Anderson *et al.* [279] employ embedded question features to highlight the most relevant image region features in visual question answering. The predicted answers are more accurately related to the question type and image content. Visual attention is widely used to learn features from two modalities.

Textual attention. Compared to visual attention, the textual attention approach is relatively less adopted. As shown in Figure 8.3(b), it has an opposite computing direction [105]. The computed weights are based on text features to obtain relevances for different image regions or objects. According to [320], the reason why textual attention is necessary is that text features from the multimodal models often lack detailed information for a given image. Meanwhile, the application of textual attention is less dominant as it is harder to capture semantic relevances between abstract text data and image data. Moreover, image data has always contained more irrelevant content for similar text. In other words, the text might describe only some parts within an image.

Co-attention. As shown in Figure 8.3(c), co-attention algorithm is viewed as a combination of visual attention and textual attention, which is an option to explore the inter-modality correlations [202, 204, 323, 329]. Co-attention is a particular case of joint feature embedding in which image and text features are usually treated symmetrically. Co-attention in a bi-directional way is beneficial for spatial-semantic learning. As an example, Nguyen *et al.* [329] introduce a dense symmetric co-attention method to improve the fusion performance of image and text representations for VQA. In their method, features are sampled densely to fully consider each interaction between any word in question and any image region. Meanwhile, several other works explore different formations of co-attention. Integrating image feature with hierarchical text features may vary dramatically so that the complex correlations are not fully captured. For this, Yu *et al.* [331] develop the co-attention mechanism into a generalized Multi-modal Factorized High-order pooling (MFH) block in an asymmetrical way. Thereby, higher-order correlations of multi-modal feature achieve more discriminative image-question representation and further result in significant improvement on the VQA performance.

Self-attention. Compared to the co-attention algorithm, self-attention, which considers the intra-modality relations, is less popular in deep multimodal learning. As intra-modality relation is complementary to the inter-modality relation, its exploration is considered improving the feature learning capability of deep networks. For example, in the VQA task, the correct answers are not only based on their associated words/phrases but can also be inferred from related regions or objects in an image. Based on this observation, a self-attention algorithm is proposed for multimodal learning to enhance the complementary between intra-modality relations and the inter-modality relations [332]. Self-attention has been used in different ways. For example, Gao *et al.* [333] combine the attentive vectors form self-attention with co-attention using element-wise product. Thereby the inter- and intra-modality information flow are modeled by the linear method.

It is important to note that when these four types of attention mechanisms are applied, they can be used to highlight the relevances between different image region features and word-level, phrase-level or sentence-level text features. These different

cases just need region/object proposal networks and sentence parsers. When multi-level attended features are concatenated, the final features are more beneficial for content understanding in multimodal learning.

As for single-hop and multiple-hop (stacked) attention, the difference lies in whether the attention "layer" will be used one or more times. The four mentioned attention algorithms can be applied in a single-hop manner where the relevance weights between image and text features are computed once only. However, for multiple-hop scenarios, the attention algorithm is adopted hierarchically to perform coarse-to-fine feature learning, that is, in a stacked way [202, 323, 325, 328]. For example, Xu *et al.* [325] introduce two-hop spatial attention learning for VQA. The first hop focuses on the whole and the second one focuses on individual words and produces word-level features. Singh *et al.* [328] achieve marginal improvements using "attention on attention" framework in which the attention module is stacked in parallel and for image and text feature learning. Nevertheless, a stacked architecture has tendency for gradient vanishing [323]. Regarding this, Fan *et al.* [323] propose stacked latent attention for VQA. Particularly, all spatial configuration information contained in the intermediate reasoning process is retained in a pathway of convolutional layers so that the vanishing gradient problem is tackled.

In summary, to better understand the content in visual and textual modality, attention mechanisms provide a pathway for aligning the multimodal semantic correlations. With different multimodal applications, attention mechanisms (single-hop or multiple-hop) can have different benefits. To this end, we briefly make a comparison for single-hop and multiple-hop with respect to their advantages, disadvantages, and the applicable scenarios in Table 8.1.

Table 8.1: Brief comparisons of two attention categories

Hop(s)	Advantages	Disadvantages	Applicable scenarios
Single	More straightforward and training effective since the visual-textual interaction occurs a single time	Less focused on complex relations between words. Insufficient to locate words or features on complicated sentences	No explicit constraints for visual attention. Suitable for capturing relations in short sentences as tends to be paid much to the most frequently words.
Multiple	More sophisticated and accurate, especially for complicated sentences. Each iteration provides newly relevant information to discover more fine-grained correlations between image and text.	Less training effective due to re-assigning attention weights multiple times. Sharing structures and parameters leads to attention bias (similar attention weights in all hops). Might suffer from the gradient vanishing problem [323].	Beneficial for multimodal learning involved long sentences. More suitable for sentence embedding in text classification or machine translation tasks. Beneficial for combining with memory networks due to the repeatedly or iteratively information extraction process.

8.3.3 Common latent space learning

As illustrated in Figure 8.1, unimodal features distribute inconsistently and are not directly comparable. It is necessary to further map these unimodal features into a common latent space with the help of an embedding networks (*e.g.* MLP). Due to this, common latent feature learning has been a critical procedure for exploiting multimodal correlations. In the past years, various constraint and regularization

methods have been introduced into multimodal applications. In this section, we include these ideas, such as attention mechanisms, which aim to retain similarities between unimodal image and text features.

According to [334], multimodal feature learning methods include joint and coordinated methods. The joint feature embeddings are defined as:

$$J = \mathcal{J}(x_1, ..., x_n, y_1, ..., y_n) \tag{8.1}$$

while coordinated feature embeddings are represented as:

$$F = \mathcal{F}(x_1, ..., x_n) \sim \mathcal{G}(y_1, ..., y_n) = G \tag{8.2}$$

where J refers to the jointly embedded features, F and G denote the coordinated features. $x_1, ..., x_n$ and $y_1, ..., y_n$ are n-dimension unimodal feature representations from two modalities. The mapping functions $\mathcal{J}(\cdot)$, $\mathcal{F}(\cdot)$, and $\mathcal{G}(\cdot)$ denote the deep networks to be learned, "$\sim$" indicates that the two unimodal features are separated but are related by some similarity constraints.

a. Joint feature embedding

In deep multimodal learning, joint feature embedding is a straightforward way in which unimodal features are combined into the same presentation. The fused features are used to make a classification in cross-modal retrieval. It also can be used for performing sentence generation in VQA [279, 304].

In early studies, some basic methods are employed for joint feature embedding such as feature summation, feature concatenation [23, 291, 292], and element-wise inner product [324, 329], the resultant features are then fed into a multi-layer perceptron to predict similarity scores. These approaches construct a common latent space for features from different modalities but cannot preserve their similarities while fully understanding the multimodal content. Alternatively, more complicated bilinear pooling methods such as Multimodal Compact Bilinear (MCB) pooling [335]. However, the performance of MCB is based on a higher-dimensional space. Regarding this demerit, Multimodal Low-rank Bilinear pooling [336] and Multimodal Factorized Bilinear pooling [331] are proposed to overcome the high computational complexity when learning joint feature. Moreover, Hedi *et al.* [337] introduce a tensor-based Tucker decomposition strategy, MUTAN, to efficiently parameterized bilinear interactions between visual and textual representations so that the model complexity is controlled and the model size is tractable. In general, to train an optimal model to understand semantic correlations, classification-based objective functions [313] and regression-based objective functions [23, 292] are commonly adopted.

Bilinear pooling methods are based on outer products to explore correlations of multimodal features. Alternatively, neural networks are used for jointly embedding features since its learnable ability for modelling the complicated interactions between

image and text. For instance, auto-encoder methods, as shown in Figure 8.2(b), are used to project image and text features with a shared multi-layer perceptron (MLP). The similar multimodal transformer introduced in [332] constructs a unified joint space for image and text. In addition, sequential networks are also adopted for the latent space construction. Take visual question answering as an example, based on the widely-used "encoder-decoder" framework, image features extracted from the encoder are fed into the decoder (*i.e.* RNNs [310]), and finally combined with text features to predict correct answers [302, 312, 326]. There are several ways to combine features. Image features can be viewed as the first "word" and concatenate all real word embeddings from the sentences. Alternatively, image features can be concatenated with each word embedding then fed them into RNNs for likelihood estimation. Considering the gradient vanishing in RNNs, CNNs are used to explore complicated relations between features [203, 338]. For example, convolutional kernels are initialized under the guidance of text features. Then, these text-guided kernels operate on extracted image features to maintain semantic correlations [338].

The attention mechanisms in Section 8.3.2 can also be regarded as a kind of joint feature alignment method and are widely used for common latent space learning. Theoretically, these feature alignment schemes aim at finding relationships and correspondences between instances from visual and textual modalities [320, 334]. In particular, the mentioned co-attention mechanism is a case of joint feature embedding in which image and text features are usually treated symmetrically. The attended multimodal features are beneficial for understanding the inter-modality correlations. Attention mechanisms for common latent space learning can be applied in different formations, including bi-directional [329], hierarchical [331], and stacked [202, 325]. More importantly, the metrics for measuring similarity are crucial in attentive importance estimation. For example, the importance estimation by simple linear operation may fail to capture the complex correlations between visual and textual modality while the Multi-modal Factorized High-order pooling (MFH) method can learn higher-order semantic correlations and achieve marginal performance.

To sum up, joint feature embedding methods are basic and straightforward ways to allow learning interactions and perform inference over multimodal features. Thus, joint feature embedding methods are more suitable for situations where image and text raw data are available during inference, and joint feature embedding methods can be expanded into situations when more than two modalities are present. However, for content understanding among inconsistently distributed features, as reported in previous work [302], there is potential for improvement in the embedding space.

b. Coordinated feature embedding

Instead of embedding features jointly into a common space, an alternative method is to embed them separately but with some constraints on features according to their

similarity (*i.e.* coordinated embedding). For example, the above-noted reconstruction loss in auto-encoders can be used to constraining multimodal feature learning in the common space. Using traditional canonical correlation analysis, as an alternative, the correlations between two kinds of features can be measured and then maintained. To explore semantic correlation in a coordinated way, generally, there are two commonly used categories: classification-based methods and verification-based methods.

For classification-based methods when class label information is available, these projected image and text features in the common latent space are used for label prediction [177, 178]. Cross-entropy loss between the inference labels and the ground-truth labels is computed to optimize the deep networks, see Figure 8.1, via the back-propagation algorithm. For classification-based methods, class labels or instance labels are needed. They map each image feature and text feature into a common space and guarantee the semantic correlations between two types of features. Classification-based methods mainly concern the image-text pair with the same class label. For the image and unmatched text (vice versa), classification-based methods have less constraints.

Different from classification-based methods, the verification-based methods can constrain both the matched image-text pairs (similar or have the same class labels) and unmatched pairs (dissimilar or have the different class labels). Verification-based methods are based on metric learning among multimodal features. Given similar/dissimilar supervisory information between image and text, these projected multimodal features should be mapped based on their corresponding similar/dissimilar information. In principle, the goal of the deep networks is to make similar image-text features close to each other while mapped dissimilar image-text features further away from each other. Verification-based methods include pair-wise constraint and triplet constraint, both of which form different objective functions.

For pair-wise constraint, the key point lies in constructing an inference function to infer similarity of features. For example, Li *et al.* [178] construct a Bayesian network, rather than a simple linear operation, to preserve the similarity relationship of image-text pairs. In addition, triplet constraint is also widely used for building the common latent space. Typically, bi-directional triplet loss function is applied to learn feature relevances between two modalities [177, 339]. Inter-modality correlations are learned well when triplet samples interchange within image and text. However, a complete deep multimodal model should also be able to capture intra-modality similarity, which is a complementary part for inter-modality correlation. Therefore, several works consider combining intra-modal triplet loss in feature learning in which all triplet samples are from the same modality (*i.e.* image or text data).

These classification-based and verification-based approaches are widely used for deep multimodal learning. Although the verification-based methods overcome some limits of classification-based methods, they still face some disadvantages such as the

negative samples and margin selection, which inherit from metric learning [186]. Recently, new ideas on coordinated feature embedding methods have combined adversarial learning, reinforcement learning, cycle-consistent constraints to pursue high performance.

Combined with adversarial learning. Classification- and verification-based methods focus on the semantic relevance between similar/dissimilar pairs. Adversarial learning focuses on the overall distributions of two different modalities instead of just focusing on each pair. The primary idea in GANs is to determine whether the input image-text pairs are matched [287, 288, 300].

In new ideas of adversarial learning for multimodal learning, an implicit generator and a discriminator are designed with competitively goals (*i.e.* the generator enforces similar image-text features be close while the discriminator separates them into two clusters). Therefore, the aim of adversarial learning is not to make a binary classification ("True/False"), but to train two groups of objective functions adversarially, it will enable the deep networks with powerful ability and focus on holistic features. For example, in recent works [177, 178, 309], a modality classifier is constructed to distinguish the visual modality and textual modality according to the input multimodal features. This classifier is trained adversarially with other sub-networks which constrain similar image-text feature to be close. Furthermore, adversarial learning is also combined with a self-attention mechanism to obtain attended regions and unattended regions. This idea is imposed on the formation of a bi-directional triplet loss to perform cross-modal retrieval.

Combined with reinforcement learning. Reinforcement learning has been incorporated into deep network structures (*e.g.* encoder-decoder framework) for image captioning [272, 273], VQA [340] and cross-modal retrieval. Because reinforcement learning avoids exposure bias [273, 339] and non-differentiable metric issue [272, 339]. It is adopted to promote multimodal correlation modeling. To incorporate reinforcement learning, its basic components are defined (*i.e.* "agent", "environment", "action", "state", and "reward"). Usually, the deep models such as CNNs or RNNs are viewed as the "agent", which interacts with an external "environment" (*i.e.* text features and image features), while the "action" is the prediction probabilities or words of the deep models, which influence the internal "state" of the deep models (*i.e.* the weights and bias). The "agent" observes a "reward" to motivate the training process. The "reward" is an evaluation value through measuring the difference between the predictive distribution and ground-truth distribution. For example, the "reward" in image captioning is computed from the CIDEr (Consensus-based Image Description Evaluation) score of a generated sentence and a true descriptive sentence. The "reward" plays an important role for adjusting the goal of predictive distribution towards the ground-truth distribution.

Reinforcement learning is commonly used in generative models in which image patch features or word-level features are regarded as sequential inputs. When incorporating reinforcement learning into deep multimodal learning, it is important to define an algorithm to compute the expected gradients and the "reward" as a reasonable optimization goal.

For the first term, the expected gradients, REINFORCE algorithm [341] is widely used as a policy gradient method to compute gradients, then to update these "states" via back-propagation algorithms [340, 342]. For the second term, there are several different alternatives. For example, the difference, evaluated by the popular metric CIDEr, between the generated captions and true description sentences in image captioning is used as a "reward" [272, 342]. Instead of measuring the difference, sample similarity is more straightforward to track. As an example, visual-textual similarity is used as "reward" after deep networks are trained under the ranking loss (*e.g.* a triplet loss) [339]. The design of triplet ranking loss function is diverse, such as in a bi-directional manner or based on inter-modal triplet sampling [339].

Combined with cycle-consistent constraint. Class label information or relevance information between image and text is crucial for understanding semantic content. However, this supervisory information sometimes is not available for training deep networks. In this case, a cycle-consistent constraint is employed for unpaired image-text inputs. The basic idea of a cycle-consistent constraint is dual learning in which a closed translation loop is used to regularize the training process. This self-consistency constraint allows a predictive distribution to retain most of the correlations of the original distribution to improve the stability of network training. In principle, a cycle-consistent constraint includes a forward cycle and backward cycle. The former relies on the loss function $F(G(X)) \approx X$, while the latter relies on another loss function $G(F(Y)) \approx Y$. In these two functions, $F(\cdot)$ is a mapping process from Y to X and $G(\cdot)$ is a reversed process from X to Y. Cycle-consistency has been used on several tasks such as cross-modal retrieval [269], image generation [268], and VQA [343].

Cycle-consistency is an unsupervised learning method for exploring semantic correlation in the common latent space. To ensure predictive distribution and retain as many correlations as possible, the aforementioned forward and backward cycle-consistent objective functions are necessary. The feature reconstruction loss function acts as the cycle-consistency objective function. For example, Gorti *et al.* [268] utilize the cross-entropy loss between generated words and the actual words as cycle-consistency loss values to optimize the process text-to-image-to-text translation. For cross-modal retrieval tasks, Li *et al.* [344] adopt Euclidean distance between predictive features and reconstructed features as the cycle-consistency loss where the two cycle loss functions interact in a coupled manner to produce reliable codes.

Currently, the application of cycle-consistent constraints for deep multimodal learning can be categorized as structure-oriented and task-oriented. The former group

focuses on making several components in a whole network into a close loop in which output of each component is used as the input for another component. Differently, task-oriented group concerns to exploit the complementary relations between tasks. Thus, there are two independent tasks (*e.g.* VQA and VQG) in the close loop.

For structure-oriented groups, the cycle-consistent idea is combined with some popular deep networks, such as GANs, to make some specific combinations. In these methods, image features are projected as "text features" and then reconstructed back to itself. Currently, the combination with GANs is a popular option since paired correspondence of modalities can be learned in the absence of a certain modality (*i.e.* via generation). For example, Wu *et al.* [269] plug a cycle-consistent constraint into feature projection between image and text. The inversed feature-learning process is constrained using the least absolute deviation. The whole process is just to learn a couple of generative hash functions through the cycle-consistent adversarial learning. For this limit, Li *et al.* [344] devise an outer-cycle (for feature representation) and an inner-cycle (for hash code learning) constraint to combine GANs for cross-modal retrieval. Thereby, the objects for which the cycle-consistency loss constrains have increased. Moreover, in their method, the discriminator should distinguish if the input feature is original (viewed as *True*) or generated (viewed as *False*).

For task-oriented groups, cycle-consistency is adopted into dual tasks. In cycle-consistency, we use an inverse process (*task A* to *task B* to *task A*) to improve the results. When a whole network performs both tasks well, it indicates that the learned features between the tasks have captured the semantic correlations of two modalities. For example, Li *et al.* [343] combine visual question answering (VQA) and visual question generation (VQG), in which the predicted answer is more accurate through combining image content to predict the question. In the end, the complementary relations between questions and answers lead to performance gains. For text-image translation, a captioning network is used to produce a caption which corresponds to a generated image from a sentence using GANs [268]. The distances between the ground truth sentences and the generated captions are exploited to improve the network further. The inverse translation is beneficial for understanding text context and the synthesized images. To sum up, there are still some questions to be explored in task-oriented ideas, such as the model parameter sharing scheme, and these implicit problems make the model more difficult to train and might encounter gradient vanishing problems, the task-oriented cycle-consistent constraint is applied to unify multi-task applications into a whole framework and attracts more research attention.

8.4 Results and Discussions

The aforementioned ideas have made some progress on various multimodal tasks. For example, for cross-modal retrieval, we presented the achieved progress and state-

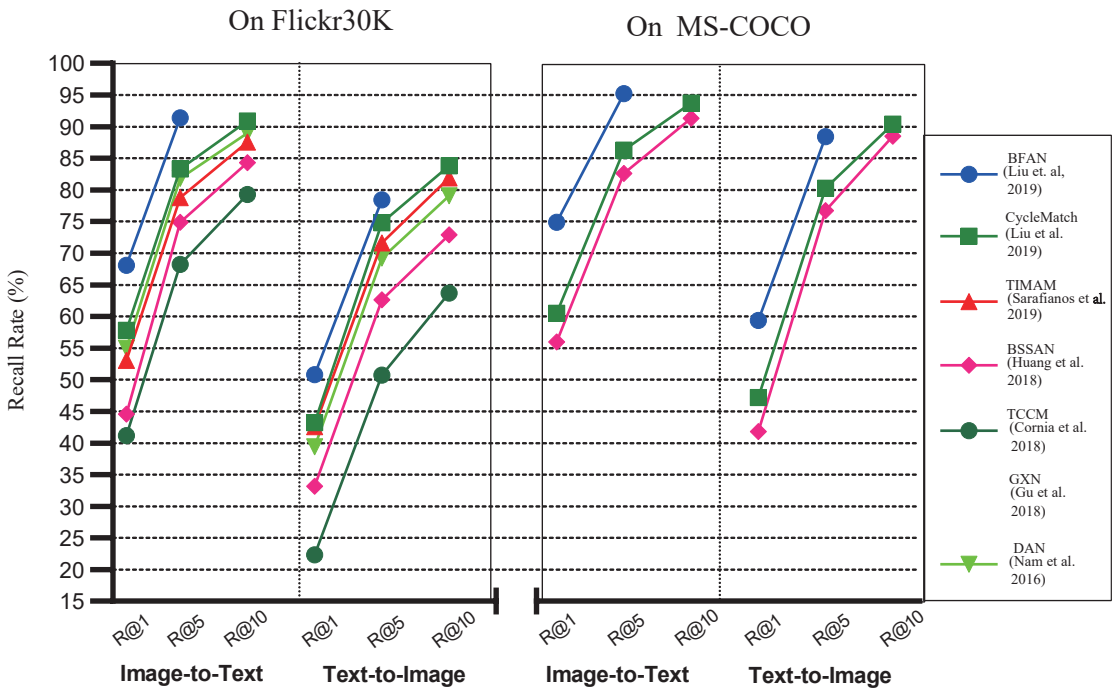


Figure 8.4: The achieved progress of cross-modal retrieval on the Flickr30K [345] and the MS-COCO [192] datasets.

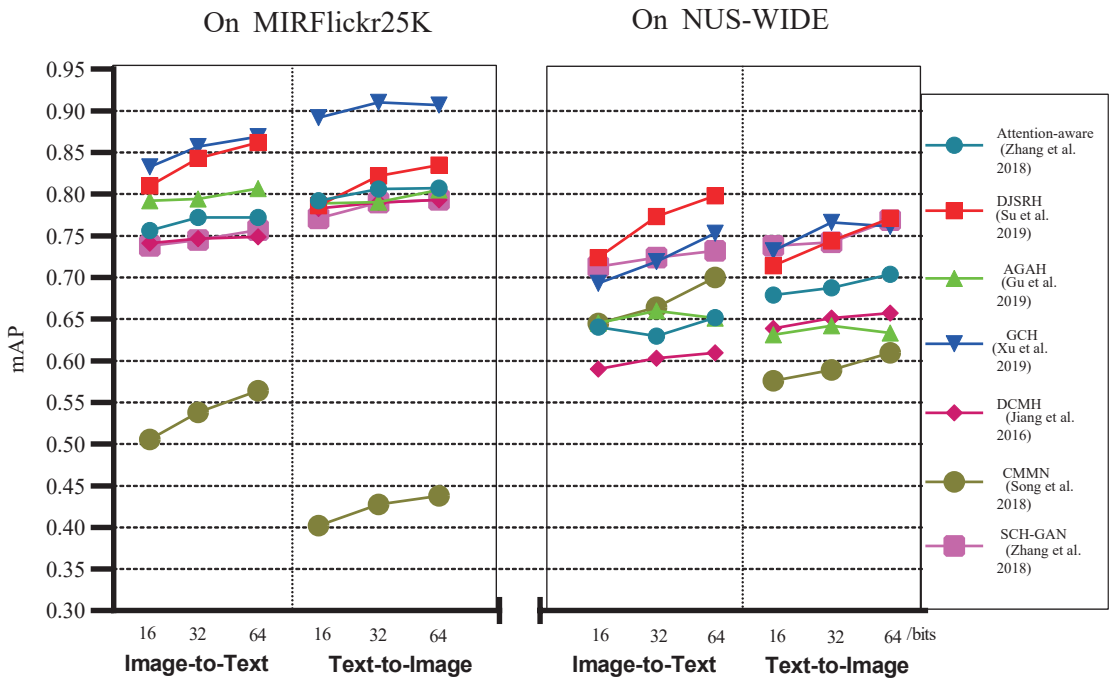


Figure 8.5: The achieved progress of cross-modal hash retrieval on the MIRFlickr25k [182] and the NUS-WIDE [183] datasets. Hashing methods have higher retrieval efficiency using the binary hash codes.

of-the-art of recent methods on the Flickr30K [345] and the MS-COCO [192] datasets in Figure 8.4. For hashing retrieval methods, we presented the achievement on the MIRFlickr25k [182] and the NUS-WIDE [183] datasets in Figure 8.5. As we can

see from these statistics, the progress is notable in recall rate (*i.e.* the fraction of queries for which the top K nearest neighbors are retrieved correctly) and mAP (*i.e.* the mean of the average precision scores for each query) in cross-modal retrieval. As can be seen from the results, there is still room for improvement in the current limitations of multimodal content understanding. In terms of other three tasks, (*i.e.* image generation, image captioning and VQA), the achieved performance in recent years are reported in Tables 8.2, 8.3, and 8.4, respectively.

Multi-task integrated networks might be helpful and complementary for content understanding as different applications capture semantic correlations from different perspectives. Effort has been made on integrating image captioning and cross-modal retrieval tasks, image captioning and visual question answering, image generation and image retrieval. Nevertheless, these combined applications are only based on two modalities. Considering the complementary characteristic among modalities (conveying the same concept), it might be promising to fuse more than two modalities to enable machines to understand their semantic correlations. Undoubtedly, it will be more challenging for aligning these diverse data. There are some explorations in this direction. Aytar *et al.* [367] present a deep cross-modal convolutional network to learn a representation that is aligned across three modalities: sound, image, and text. The network is only trained with "image + text" and "image + sound" pairs. He *et al.* [368] construct a new benchmark for cross-media retrieval in which image, text, video, and audio are included. It is the first benchmark with 4 media types for fine-grained cross-media retrieval. However, this direction is still far from satisfactory.

Deep neural networks, including convolutional neural networks and recurrent neural networks, have made the unimodal feature extraction and multimodal feature learning end-to-end trainable. The representations from multimodal data can be automatically learned effectively, without the need of requiring expert knowledge in a certain field, which makes the process of understanding of multimodal content more intelligent. However, the disadvantages of deep networks for multimodal learning are obvious. It is well-known that the deep networks depend on a massive of multiple-modality data to train, but the less biased datasets are not so common. More importantly, deep networks for multimodal learning lacks of interpretability to some extent. Although joint embedding or coordinated embeddings methods can be utilized, it still needs to figure out which modality (or its features) plays more important role for the final content understanding.

From a technical viewpoint, graph-based networks are an important direction for future research. Currently, graph representation is constructed within intra-modality to present the semantic relations, which can be further explored in the future. Meanwhile, the exploration of graph-based networks can be deepened by examining scalability and heterogeneity. Finally, generation-based tasks such as image generation and image captioning are effective for unsupervised learning, since numerous labeled

Table 8.2: Performance of image captioning on the MS-COCO dataset [192]

Methods	CIDEr		ROUGE-L		METEOR		BLEU1		BLEU2		BLEU3		BLEU4		KeyNotes
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	
StructCap [318]	94.3	95.8	53.5	68.2	25.0	33.5	73.1	90.0	56.5	81.5	42.4	70.9	31.6	59.9	Attention + VP-tree for visual features
Semantic-Attn [346]	95.3	94.8	53.4	68.4	25.1	34.0	72.4	90.7	55.8	82.2	42.3	71.7	32.0	60.7	Visual attention
CGAN [276]	102.0	-	52.7	-	24.8	-	-	-	-	-	39.3	-	29.9	-	CGAN + Reinforcement learning
Adaptive-Attn [347]	104.2	105.9	55.0	70.5	26.4	35.9	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	Adaptive visual attention
SCST [272]	114.7	116.7	56.3	70.7	27.0	35.5	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	Reinforcement learning
RL-GAN [348]	-	-	-	-	24.3	-	71.6	-	51.8	-	37.1	-	26.5	-	GAN + Reinforcement learning
SOT [266]	106.1	108.7	55.5	69.9	25.9	34.2	78.7	93.5	61.5	85.5	46.5	74.8	34.5	63.3	Visual attention
SR-PL [339]	117.1	-	57.0	-	27.4	-	80.1	-	63.1	-	48.0	-	35.8	-	Reinforcement learning
Up-Down [272]	117.9	120.5	57.1	72.4	27.6	36.7	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	Top-down and bottom-up attention
CAVP [273]	121.6	123.8	58.2	73.1	28.1	37.0	80.1	94.9	64.7	88.8	50.0	79.7	37.9	69.0	Reinforcement learning
RFNet [311]	122.9	125.1	58.2	73.1	28.2	37.2	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	Visual attention
iVQA [349]	168.2	-	46.6	-	20.1	-	42.1	-	32.0	-	25.3	-	20.5	-	Reinforcement learning
UnsupervisedIC [316]	54.9	-	43.1	-	17.9	-	58.9	-	40.3	-	27.0	-	19.6	-	VAE + GAN (unsupervised)
Graph-align [314]	69.5	-	-	-	20.9	-	67.1	-	47.8	-	32.3	-	21.5	-	VAE + Graph embed + cycle (unpaired)
Self-critical [342]	112.6	115.3	56.1	70.4	26.9	35.4	77.6	93.1	61.3	86.1	46.5	76.0	34.8	64.6	Reinforcement learning
PAGNet [280]	118.6	-	58.6	-	30.4	-	83.2	-	62.8	-	46.3	-	40.8	-	Attention + Reinforcement learning
RL-CGAN [312]	123.1	124.3	59.0	74.4	28.7	38.2	81.9	95.6	66.3	90.1	51.7	81.7	39.6	71.5	GAN + Reinforcement learning
SGAE [271]	123.8	126.5	58.6	73.6	28.2	37.2	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	VAE + graph embedding

Table 8.3: Performance of image generation

Methods	Caltech-UCSD Birds200		Oxford Flowers102		MS-COCO		KeyNotes
	IS	FID	IS	FID	IS	FID	
GAN-INT-CLS [22]	2.88±.04	-	2.66±.03	-	7.88±0.07	-	Vanilla GAN for image generation
TAC-GAN [313]	-	-	3.45±.05	-	-	-	Discriminator learns class information
GAWWN [293]	3.60±.07	-	-	-	-	-	Conditional objection location is learned
StackGAN [23]	3.70±.04	51.89	3.20±.01	55.28	8.45±.03	74.05	GAN in a stacked structure
StackGAN++ [291]	4.04±.05	15.3	3.26±.01	48.68	8.30±.10	81.59	GAN in a tree-like structure
HDGAN [289]	4.15±.05	-	3.45±.07	-	11.86±0.18	-	GAN in a hierarchically-nested structure
AttnGAN [292]	4.36±.03	-	-	-	25.89±.47	-	Attentional generative network
Scene graphs [287]	-	-	-	-	7.3±0.1	-	Graph convolution for graphs from text
Obj-GANs [350]	-	-	-	-	27.37±0.22	25.85	Attentive generator and object-wise discriminator
vmCAN [296]	-	-	-	-	10.36±0.17	-	Visual-memory method in GAN
AAAE [315]	-	-	-	103.46	-	-	Auto-encoders + GAN for adversarial approximation
Text-SeGAN [267]	-	-	4.03±0.07	-	-	-	Semantic relevance matching in GAN
DAI [351]	3.58±0.05	18.41±1.07	2.90±0.03	37.94±0.39	8.94±0.2	27.07±2.55	Dual inference mechanism disentangled variables
C4Synth [352]	4.07±0.13	-	3.52±0.15	-	-	-	Image generation using multiple captions
PPAN [290]	4.35±.05	-	3.53±.02	-	-	-	GAN in perceptual pyramid structure
MirrorGAN [353]	4.56±0.05	-	-	-	26.47±0.41	-	Task-oriented cycle consistency + attention
ControlGAN [354]	4.58±0.09	-	-	-	24.06±0.6	-	Attention + region-wise attribute generation
SD-GAN [355]	4.67±0.09	-	-	-	35.69±0.5	-	Disentangling high-/low-level semantics in GAN

‡ To evaluate the identification and diverse of generated image, Inception Score (IS) and Fréchett Inception Distance (FID) are commonly used. For Inception Score, higher is better. For Fréchet Inception Distance, lower is better.

Table 8.4: Performance of visual question answering on VQA 1.0 dataset [356]

Methods	Open-ended test-std	Open-ended test-dev	MC test-std	MC test-dev	KeyNotes
Smem-VQA [325]	58.24	57.99	-	-	Spatial memory network stores image region features
DMN+ [324]	60.4	60.3	-	-	Improved dynamic memory network for VQA
MLB [336]	65.07	64.89	68.89	-	Low-rank bilinear pooling for similarity learning
MCB [335]	66.5	66.7	70.1	70.2	Multimodal compact bilinear pooling for similarity learning
High-order Attn [357]	-	-	69.3	69.4	Attention mechanisms learn high-order feature correlations
DAN [202]	64.2	64.3	69	69.1	Co-attention networks for multimodal feature learning
MLAN [358]	65.3	65.2	70	70	Multi-level co-attention for feature alignment
SVA [359]	66.1	66	-	-	Visual attention on grid-structured image region feature learning
MFB [331]	66.6	66.9	71.4	71.3	Multi-modal factorized bilinear pooling for similarity learning
MUTAN [337]	67.36	67.42	-	-	Multimodal tucker fusion for similarity learning
Graph VQA [319]	70.42	-	74.37	-	Graph representation for scene and question feature learning
MAN-VQA [326]	64.1	63.8	69.4	69.5	Memory-augmented network for feature learning and matching
QGHC [338]	65.9	65.89	-	-	Question-guided convolution for visual-textual correlations learning
Dual-MFA [360]	66.09	66.01	69.97	70.04	Co-attention for visual-textual feature learning
VKMN [321]	66.1	66	69.1	69.1	Visual knowledge memory network for feature learning
CVA [361]	66.2	65.92	70.41	70.3	Cubic visual attention for object-region feature learning
DCN [329]	67.02	66.89	-	-	Dense co-attention for feature fusion
DRAU [362]	67.16	66.86	-	-	Recurrent co-attention for feature learning
ODA [327]	67.97	67.83	72.23	72.28	Object-difference visual attention to fuse features
ALARR [363]	68.43	68.61	71.28	68.43	Adversarial learning for pair-wise feature discrimination
DF [364]	68.48	68.62	73.05	73.31	Differential network for visual-question feature learning
Relational Encoding [365]	69.3	69.1	-	-	Textual attention for question feature encoding
DCAF [366]	70.0	69.9	-	-	Dense co-attention for feature fusion

training data can be generated from the deep networks. Combined with reinforcement learning, the image generation process is more controllable. For example, some fine-grained attributes including texture, shape and color can be specified during deep network training. Once it understands the content between modalities, the deep network, like an agent, will synthesize photo-realistic images, which can be used in other applications.

8.5 Chapter Conclusions

In this chapter, we have conducted a review of recent ideas and trends in deep multimodal learning (image and text) including popular structures and algorithms. We analyzed two major challenges in deep multimodal learning for which these popular structures and algorithms target. Specifically, popular structures including autoencoders, generative adversarial nets and their variants perform uni-directional and bi-directional multimodal tasks. Based on these popular structures, we introduced current ideas about multimodal feature extraction and common latent feature learning which plays crucial roles for better content understanding within a visual and textual modality. For multimodal feature extraction, we introduced graph convolutional networks and memory-augmented networks. For common latent feature learning, we presented the joint and coordinated feature embedding methods including the recently proposed objective functions.

Chapter 9

Conclusions

Finding information in digital datasets and libraries is one of the grand challenges of our generation. Finding images or image retrieval is a major sub-problem and has recently had significant advances due to the groundbreaking developments in deep visual learning. In this thesis, we have explored and designed algorithms for retrieval tasks via deep learning methods, including unimodal image retrieval and cross-modal retrieval.

In Chapter 2, we presented a comprehensive review on deep learning for image retrieval. We introduced the popular backbone deep network architectures, that is widely used for extracting retrieval feature representations, and summarized three aspects of the challenge for deep image retrieval, including (1) reducing the semantic gap, (2) improving retrieval scalability, and (3) balancing retrieval accuracy and efficiency. Based on these main challenges, we presented methodologies for retrieval, including feature extraction, feature fusion, and feature enhancement methods. These methods can be employed in off-the-shelf convolutional neural networks. Also, they can be applied when deep networks are fine-tuned on the new target retrieval datasets. We analyzed supervised and unsupervised fine-tuning methods for the updating of network parameters. For these methodologies, we compared their performance on four retrieval benchmarks. This chapter aims to give a global view of intelligent image retrieval.

Finding an image of interest may require searching through thousands, millions, or even billions of images. Therefore, searching efficiently is as critical as searching accurately. To enable accurate and efficient retrieval of massive image collections, learning compact and rich feature representations is critical. In Chapter 3, we focus on cross-modal hash retrieval because hash code learning has high efficiency in computation and storage. We proposed an information entropy loss function based on Shannon information theory to reduce the heterogeneity gap, and thereby build a better common space to align the visual and textual modalities. We regularized real-valued features and the binary hash codes using the proposed information entropy loss. As demonstrated in Chapter 3, the challenge of performing cross-modal retrieval lies in how to measure the semantic similarity between data from different modalities. For this purpose, in Chapter 4, we proposed to integrate information theory and adversarial learning to learning the cross-modal features. Combining information theory and adversarial learning is beneficial in discovering the distribution differences between modalities to minimize the heterogeneity gap and enable more accurate retrieval. To guarantee the semantic similarity between data from visual and textual modalities, we adopted a bi-directional ranking loss function and a cross-modal feature projection method. Moreover, we adopted the Kullback–Leibler divergence to address the data imbalance issue which exists in the cross-modal datasets where each image is described by five sentences. The proposed method is evaluated by thorough experimental results on four well-known datasets using four deep models.

In Chapters 3 and 4, the proposed methods were trained and evaluated on fixed datasets. In Chapter 5, we explored fine-grained image retrieval in the context of incremental learning where deep networks are trained by using new data only. The new data is added at once or sequentially into the existing old data, where we employed the knowledge distillation method, which is computed based on the output probabilities from the final classifier, and the maximum mean discrepancy loss, which is based on the retrieval feature representations from the intermediate layer. The proposed method was compared with the state-of-the-art methods and to show its efficacy. We also applied the proposed method for incremental image classification tasks.

In Chapter 6, we further explored methods for incremental fine-grained image retrieval. Previously, in Chapter 5, we only used the penultimate model as the teacher model to regularize the current student model which learns on the new task. As incremental learning proceeds, especially when new data are added sequentially, knowledge distillation based on the stream of models will be memory-consuming and make the learning complex. We proposed a feature estimation method to estimate representative features from the models trained on earlier old tasks so that saving this model stream is unnecessary. Quantitative and qualitative experiments on two common benchmarks demonstrate that the proposed approach is effective for achieving optimal performance on both the old and new tasks when new incoming data are added at once or sequentially.

In Chapter 7, we explored fine-grained image retrieval in a lifelong manner. In contrast to Chapter 6 and Chapter 5, the images in the newly added data are semantically different from the ones in the already trained data. These semantic drifts make minimizing the forgetting ratio on previous tasks more difficult. In addition, we considered improving the generalization ability of the trained networks on the new tasks. To this end, we proposed a dual knowledge distillation framework that includes two professional teachers and a self-motivated student. To further alleviate the forgetting issue, we used the stored running statistics of the BatchNorm layers of the frozen teacher to generate several representative images. We evaluated the proposed framework on three benchmarks, where the scenarios of two-task sequence and three-task sequence are considered.

In Chapter 8, we presented four popular multimodal applications, including cross-modal retrieval, image captioning, image generation, and visual question answering. We introduced recent new ideas and trends of these applications from the viewpoint of structure for multimodal feature extraction and the strategies for multimodal feature learning. These novel ideas are important for better multimodal content understanding and can be further used to improve performance in intelligent image retrieval.

9.1 Limitations and Possible Solutions

Although our research has reached its aims, there still exist some limitations for our initial explorations for intelligent image retrieval.

First, in Chapter 4, we explored integrating information theory and adversarial learning for cross-modal retrieval, in which the information entropy loss was computed only based on image modality and text modality. Therefore, the feature vectors extracted from these two modalities are projected into a common feature space but the associations and alignments between cross-modal features are neglected. However, retrieval performance depends on the matching of each image-text feature pair. For some large-scale datasets, each category may include a large number of image-text pairs. Thus, it is valuable to make the information entropy loss specific for each category so that the discrepancy between two modalities can be reduced more granularly.

Second, we explored image retrieval in the context of incremental learning in Chapters 5, 6, and 7, by focusing on the representations extracted from the teacher-student structure to distill correlations. Thus, both old tasks and new tasks are trained on the same representations. However, regularizing directly on the representations may be overly restrictive for the learning on the new tasks. We find the accuracy of new tasks on the CUB-Birds dataset is still lower than the upper bound of joint training (see Table 6.1). For this limitation, instead of regularizing the representations, it may be promising to project them into a sub-space using an auto-encoder or a variational auto-encoder. Afterward, informative parts of the representations for the old tasks are captured and kept unchanged, while others that are not meaningful for the old tasks allow the learning for new tasks.

Third, we proposed a feature estimation method in Chapter 6 to minimize the forgetting ratio in previous tasks. In fact, effectively estimating representations for all previous models depends on the parameter inheritance of model initialization at the start of each incremental step. However, estimated features from the penultimate model to the first one are not accurate enough due to the accumulative estimation errors. We resolved this limitation by aligning estimated features with descending importance and demonstrated its effectiveness experimentally. Nevertheless, distilling knowledge on the stream of models is worth further investigation theoretically. Sequence modeling via the recurrent network [369] may be a direction that deserves to be explored.

9.2 Future Research Directions

In terms of future work, there are several directions into which we can extend our research work:

1. Unsupervised intelligent image retrieval. We have explored intelligent image retrieval in a supervised manner. However, supervisory information such as class labels are time-consuming and labor-intensive to collect. Therefore, it is valuable to investigate unsupervised image retrieval. For example, the proposed Shannon information loss functions in Chapter 3 and Chapter 4 are label-free and can be used in some unsupervised learning scenarios. It may be more difficult for lifelong image retrieval in an unsupervised manner that uses the teacher-student framework. Without the supervisory information to regularize the training of the student network, the student network may suffer from more severe forgetting on the previous tasks. One possible solution is to employ the Variational AutoEncoder (VAE), which can be used for unsupervised learning, in lifelong representations learning.

2. Multimodal retrieval. In an information era, people can search for the item of interest by using different kinds of queries which makes the field of multimodal retrieval an area that richly deserves to be explored. One of the challenges for multimodal retrieval is to align features from different modalities in a shared latent space. We have examined the application of combining Shannon information entropy with adversarial learning for cross-modal retrieval. We find that Shannon information entropy can be used for multimodal feature learning by estimating the modality uncertainty. It will be promising to explore Shannon entropy further when applied to other kinds of cross-modal feature learning similar to image-text retrieval, such as video-text, audio-video, and audio-text matching, which aims at learning modality-invariant representations.

3. Zero-shot learning for image retrieval. The popularity of media platforms and the rapid development of novel techniques makes it very convenient for people to share their images, and as a result, the number of images on the Internet has increased exponentially where there often exist "unseen" images or categories. However, most datasets are static and offer a limited amount of objects and categories for feature learning. Thus, the retrieval algorithms or systems may suffer from the scarcity of the appropriate training data for these unseen images. Therefore, there is a need to extend conventional image retrieval methods to a zero-shot learning scenario where we can retrieve both seen and unseen categories from the system. Furthermore, combined with unsupervised methods, zero-shot learning algorithms can significantly improve the flexibility and generalization of image retrieval systems.

4. Incremental learning for image retrieval. Content-based image retrieval can be divided into category-level image retrieval and instance-level image retrieval. In our work, we have paid attention to explore category-level image retrieval in the context of lifelong learning. To avoid forgetting ratios on the already trained tasks, more techniques may be necessary, such as hierarchical learning. Since images used in the incremental fine-grained image retrieval share subtle inter-class variations and

larger intra-class variations, it is valuable to learn hierarchical domain knowledge. Furthermore, examining instance-level image retrieval in incremental learning is also promising.

5. Deploy image retrieval for practical applications. Existing image retrieval technologies are trained and evaluated on standard benchmarks, and various metric learning methods are explored for retrieval on fine-grained datasets. However, these technologies are still far from real-world applications such as face search, fashion search, person re-identification, shopping recommendation systems, or medical image retrieval. In these practical applications, the purpose of image retrieval may not just be retrieving images for general content on standard benchmarks, but also for more refined information. It is challenging to deploy image retrieval for specific scenarios. For example, as a specific instance search topic, person re-identification systems may encounter images with low-resolution or with inferior quality due to inadequate illumination. Existing techniques such as attention mechanisms and region proposal networks can be adopted to guarantee performance. In addition, it is valuable to explore multi-modal retrieval in practical applications. This means that image retrieval can also be combined with other auxiliary modalities such as words, phrases, and sentences to meet the different retrieval expectations of users.

Bibliography

[1] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1349–1380

[2] Zhang, L., Rui, Y.: Image search from thousands to billions in 20 years. ACM Transactions on Multimedia Computing, Communications, and Applications **9** (2013) 36

[3] Zheng, L., Yang, Y., Tian, Q.: SIFT meets CNN: A decade survey of instance retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2018) 1224–1244

[4] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

[5] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: European Conference on Computer Vision. (2006) 404–417

[6] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2005) 886–893

[7] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2003) 1470–1477

[8] Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Proceedings of the International Conference on Neural Information Processing Systems. (1999) 487–493

[9] Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: European Conference on Computer Vision. (2010) 143–156

[10] Li, X., Uricchio, T., Ballan, L., Bertini, M., Snoek, C.G., Bimbo, A.D.: Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. ACM Computing Surveys (CSUR) **49** (2016) 14

[11] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2009) 248–255

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems. (2012) 1097–1105

[13] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 770–778

[14] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 4700–4708

[15] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2014) 580–587

[16] Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE/CVF International Conference on

Computer Vision. (2015) 1440–1448

[17] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proceedings of the International Conference on Neural Information Processing Systems. (2015) 91–99

[18] Wang, X., Shrivastava, A., Gupta, A.: A-fast-RCNN: Hard positive generation via adversary for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017)

[19] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2018) 834–848

[20] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440

[21] Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 2481–2495

[22] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning. (2016) 1060–1069

[23] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 5907–5915

[24] Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: DRAW: A recurrent neural network for image generation. In: International Conference on Machine Learning, PMLR (2015) 1462–1471

[25] Babenko, A., Lempitsky, V.: Aggregating local deep features for image retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2015) 1269–1277

[26] Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. In: European Conference on Computer Vision. (2016) 685–701

[27] Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y., Li, J.: Deep learning for content-based image retrieval: A comprehensive study. In: Proceedings of the ACM International Conference on Multimedia. (2014) 157–166

[28] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2010) 3304–3311

[29] Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: European Conference on Computer Vision. (2014) 392–407

[30] Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2015) 53–61

[31] Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: International Conference on Learning Representations. (2015) 1–12

[32] Radenović, F., Tolias, G., Chum, O.: CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In: European Conference on Computer Vision. (2016) 3–20

[33] Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22** (1951) 79–86

[34] Chen, W., Pu, N., Liu, Y., Bakker, E.M., Lew, M.S.: Domain uncertainty based on information theory for cross-modal hash retrieval. In: IEEE International Conference on Multimedia and Expo. (2019) 43–48

[35] Chen, W., Liu, Y., Bakker, E.M., Lew, M.S.: Integrating information theory and adversarial learning for cross-modal retrieval. Pattern Recognition **117** (2021) 107983

[36] McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation. (1989) 109–165

[37] Chen, W., Liu, Y., Wang, W., Tuytelaars, T., Bakker, E.M., Lew, M.S.: On the exploration of incremental learning for fine-grained image retrieval. In: The British Machine Vision Conference. (2020) 1–10

[38] Chen, W., Liu, Y., Pu, N., Wang, W., Liu, L., Lew, M.S.: Feature estimations based correlation distillation for incremental image retrieval. IEEE Transactions on Multimedia (2021)

[39] Chen, W., Wang, W., Liu, L., Lew, M.: New ideas and trends in deep multimodal content understanding: A review. Neurocomputing (2020) 195–215

[40] Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications **2** (2006) 1–19

[41] Cao, Y., Long, M., Wang, J., Zhu, H., Wen, Q.: Deep quantization network for efficient image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2016) 3457–3463

[42] Alzu'bi, A., Amira, A., Ramzan, N.: Semantic content-based image retrieval: A comprehensive study. Journal of Visual Communication and Image Representation **32** (2015) 20–54

[43] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2014) 806–813

[44] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Proceedings of the International Conference on Neural Information Processing Systems. (2014) 3320–3328

[45] Jiménez, A., Alvarez, J.M., Giró Nieto, X.: Class-weighted convolutional features for visual instance search. In: The British Machine Vision Conference. (2017) 1–12

[46] Do, T.T., Hoang, T., Tan, D.K.L., Le, H., Nguyen, T.V., Cheung, N.M.: From selective deep convolutional features to compact binary representations for image retrieval. ACM Transactions on Multimedia Computing, Communications, and Applications **15** (2019) 1–22

[47] Xu, J., Wang, C., Qi, C., Shi, C., Xiao, B.: Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2018) 7436–7443

[48] Liu, Y., Guo, Y., Wu, S., Lew, M.S.: Deepindex for accurate and efficient image retrieval. In: Proceedings of the ACM on International Conference on Multimedia Retrieval. (2015) 43–50

[49] Wu, P., Hoi, S.C., Xia, H., Zhao, P., Wang, D., Miao, C.: Online multimodal deep similarity learning with application to image retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2013) 153–162

[50] Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: European Conference on Computer Vision. (2014) 584–599

[51] Huang, C.Q., Yang, S.M., Pan, Y., Lai, H.J.: Object-location-aware hashing for multi-label image retrieval via automatic mask learning. IEEE Transactions on Image Processing **27** (2018) 4490–4502

[52] Garcia, N., Vogiatzis, G.: Learning non-metric visual similarity for image retrieval. Image and Vision Computing **82** (2019) 18–25

[53] Ong, E.J., Husain, S., Bober, M.: Siamese network of deep fisher-vector descriptors for image retrieval. arXiv:1702.00338 (2017)

[54] Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: European Conference on Computer Vision. (2016) 241–257

[55] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture

for weakly supervised place recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 5297–5307

[56] Xu, J., Wang, C., Qi, C., Shi, C., Xiao, B.: Iterative manifold embedding layer learned by incomplete data for large-scale image retrieval. IEEE Transactions on Multimedia **21** (2018) 1551–1562

[57] Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2017) 1655–1668

[58] Liu, C., Yu, G., Volkovs, M., Chang, C., Rai, H., Ma, J., Gorti, S.K.: Guided similarity separation for image retrieval. In: Proceedings of the International Conference on Neural Information Processing Systems. (2019) 1554–1564

[59] Chang, C., Yu, G., Liu, C., Volkovs, M.: Explore-exploit graph traversal for image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 9423–9431

[60] Shen, Y., Qin, J., Chen, J., Yu, M., Liu, L., Zhu, F., Shen, F., Shao, L.: Auto-encoding twin-bottleneck hashing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2818–2827

[61] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)

[62] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 1–9

[63] Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual instance retrieval with deep convolutional networks. ITE Transactions on Media Technology and Applications (2016) 251–258

[64] Jun, H., Ko, B., Kim, Y., Kim, I., Kim, J.: Combination of multiple global descriptors for image retrieval. arXiv:1903.10663 (2019)

[65] Li, Y., Kong, X., Zheng, L., Tian, Q.: Exploiting hierarchical activations of neural network for image retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2016) 132–136

[66] Qi, C., Shi, C., Xu, J., Wang, C., Xiao, B.: Spatial weighted fisher vector for image retrieval. In: IEEE International Conference on Multimedia and Expo. (2017) 463–468

[67] Mohedano, E., McGuinness, K., Giró-i Nieto, X., O'Connor, N.E.: Saliency weighted convolutional features for instance search. In: International Conference on Content-based Multimedia Indexing. (2018) 1–6

[68] Yang, F., Li, J., Wei, S., Zheng, Q., Liu, T., Zhao, Y.: Two-stream attentive CNNs for image retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2017) 1513–1521

[69] Deng, C., Yang, E., Liu, T., Li, J., Liu, W., Tao, D.: Unsupervised semantic-preserving adversarial hashing for image search. IEEE Transactions on Image Processing **28** (2019) 4032–4044

[70] Hu, H., Wang, K., Lv, C., Wu, J., Yang, Z.: Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval. IEEE Transactions on Image Processing (2018) 739–754

[71] Deng, D., Wang, R., Wu, H., He, H., Li, Q., Luo, X.: Learning deep similarity models with focus ranking for fabric image retrieval. Image and Vision Computing **70** (2018) 11–20

[72] Zhou, K., Liu, Y., Song, J., Yan, L., Zou, F., Shen, F.: Deep self-taught hashing for image retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2015) 1215–1218

[73] Yan, K., Wang, Y., Liang, D., Huang, T., Tian, Y.: CNN vs. SIFT for image retrieval: Alternative or complementary? In: Proceedings of the ACM International Conference on

Multimedia. (2016) 407–411

[74] Wei, X.S., Luo, J.H., Wu, J., Zhou, Z.H.: Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Transactions on Image Processing **26** (2017) 2868–2881

[75] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: The British Machine Vision Conference. (2014)

[76] Piras, L., Giacinto, G.: Information fusion in content based image retrieval: A comprehensive overview. Information Fusion **37** (2017) 50–60

[77] Wang, J., Zhang, T., Sebe, N., Shen, H.T., et al.: A survey on learning to hash. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018) 769–790

[78] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2014) 1717–1724

[79] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. International Journal of Computer Vision **128** (2020) 261–318

[80] Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review **53** (2020) 5455–5516

[81] Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016) 1790–1802

[82] Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marqués, F., Giro-i Nieto, X.: Bags of local convolutional features for scalable instance search. In: Proceedings of the ACM on International Conference on Multimedia Retrieval. (2016) 327–331

[83] Sharif Razavian, A., Sullivan, J., Maki, A., Carlsson, S.: A baseline for visual instance retrieval with deep convolutional networks. In: International Conference on Learning Representations. (2015)

[84] Cao, J., Liu, L., Wang, P., Huang, Z., Shen, C., Shen, H.T.: Where to focus: Query adaptive matching for instance retrieval using convolutional feature maps. arXiv:1606.06811 (2016)

[85] Reddy Mopuri, K., Venkatesh Babu, R.: Object level deep feature pooling for compact image representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2015) 62–70

[86] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision. (2014) 391–405

[87] Mairal, J., Koniusz, P., Harchaoui, Z., Schmid, C.: Convolutional kernel networks. In: Proceedings of the International Conference on Neural Information Processing Systems. (2014) 2627–2635

[88] Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 5552–5561

[89] Salvador, A., Giró-i Nieto, X., Marqués, F., Satoh, S.: Faster R-CNN features for instance search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2016) 9–16

[90] Ng, T., Balntas, V., Tian, Y., Mikolajczyk, K.: SOLAR: Second-order loss and attention for image retrieval. In: European Conference on Computer Vision. (2020) 253–270

[91] Yu, D., Liu, Y., Pang, Y., Li, Z., Li, H.: A multi-layer deep fusion convolutional neural network for sketch based image retrieval. Neurocomputing **296** (2018) 23–32

[92] Yu, W., Yang, K., Yao, H., Sun, X., Xu, P.: Exploiting the complementary strengths of multi-layer CNN features for image retrieval. Neurocomputing **237** (2017) 235–241

[93] Shen, C., Zhou, C., Jin, Z., Chu, W., Jiang, R., Chen, Y., Hua, X.S.: Learning feature embedding with strong neural activations for fine-grained retrieval. In: Proceedings of the

ACM International Conference on Multimedia. (2017) 424–432

[94] Ding, Z., Song, L., Zhang, X., Xu, Z.: Selective deep ensemble for instance retrieval. Multimedia Tools and Applications (2018) 1–17

[95] Kim, W., Goyal, B., Chawla, K., Lee, J., Kwon, K.: Attention-based ensemble for deep metric learning. In: European Conference on Computer Vision. (2018) 736–751

[96] Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.: Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. Computers & Graphics **71** (2018) 77–87

[97] Ozaki, K., Yokoo, S.: Large-scale landmark retrieval/recognition under a noisy and diverse dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop. (2019)

[98] Xuan, H., Souvenir, R., Pless, R.: Deep randomized ensembles for metric learning. In: European Conference on Computer Vision. (2018) 723–734

[99] Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., O'Connor, N.E.: Shallow and deep convolutional networks for saliency prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 598–606

[100] Chen, B.C., Davis, L.S., Lim, S.N.: An analysis of object embeddings for image retrieval. arXiv:1905.11903 (2019)

[101] Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: International Conference on Machine Learning. (2010) 111–118

[102] Wang, F., Zhao, W.L., Ngo, C.W., Merialdo, B.: A hamming embedding kernel with informative bag-of-visual words for video semantic indexing. ACM Transactions on Multimedia Computing, Communications, and Applications **10** (2014) 1–20

[103] Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 1704–1716

[104] Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International Journal of Computer Vision **105** (2013) 222–245

[105] Li, R., Jia, J.: Visual question answering with question representation update (QRU). In: Proceedings of the International Conference on Neural Information Processing Systems. (2016) 4655–4663

[106] Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Largescale image retrieval with attentive deep local features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 3456–3465

[107] Cornia, M., Baraldi, L., Serra, G., Cucchiara, R.: Predicting human eye fixations via an LSTM-based saliency attentive model. IEEE Transactions on Image Processing **27** (2018) 5142–5154

[108] Cao, J., Huang, Z., Shen, H.T.: Local deep descriptors in bag-of-words for image retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2017) 52–58

[109] Kim, J., Yoon, S.E.: Regional attention based deep feature for image retrieval. In: The British Machine Vision Conference. (2018) 209–223

[110] Chen, B., Deng, W.: Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2750–2759

[111] Deng, C., Yang, E., Liu, T., Tao, D.: Two-stream deep hashing with class-specific centers for supervised image search. IEEE Transactions on Neural Networks and Learning Systems (2019)

[112] Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 2862–2871

[113] Kang, R., Cao, Y., Long, M., Wang, J., Yu, P.S.: Maximum-margin hamming hashing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 8252–8261

[114] Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 2064–2072

[115] Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 1556–1564

[116] Long, F., Yao, T., Dai, Q., Tian, X., Luo, J., Mei, T.: Deep domain adaptation hashing with adversarial learning. In: The International ACM SIGIR Conference on Research & Development in Information Retrieval. (2018) 725–734

[117] Cao, Y., Liu, B., Long, M., Wang, J., KLiss, M.: HashGAN: Deep learning to hash with pair conditional wasserstein GAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 1287–1296

[118] Yang, E., Liu, T., Deng, C., Liu, W., Tao, D.: DistillHash: Unsupervised deep hashing by distilling data pairs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2946–2955

[119] Carreira-Perpinán, M.A., Raziperchikolaei, R.: Hashing with binary autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 557–566

[120] Do, T.T., Le Tan, D.K., Pham, T.T., Cheung, N.M.: Simultaneous feature aggregating and hashing for large-scale image search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 6618–6627

[121] Gu, Y., Wang, S., Zhang, H., Yao, Y., Yang, W., Liu, L.: Clustering-driven unsupervised deep hashing for image retrieval. Neurocomputing **368** (2019) 114–123

[122] Song, J.: Binary generative adversarial networks for image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2017)

[123] Dizaji, K.G., Zheng, F., Nourabadi, N.S., Yang, Y., Deng, C., Huang, H.: Unsupervised deep generative adversarial hashing network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 3664–3673

[124] Erin Liong, V., Lu, J., Wang, G., Moulin, P., Zhou, J.: Deep hashing for compact binary codes learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 2475–2483

[125] Cakir, F., He, K., Bargal, S.A., Sclaroff, S.: Hashing with mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2019) 2424–2437

[126] Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European Conference on Computer Vision. (2008) 304–317

[127] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2007) 1–8

[128] Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 403–412

[129] Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep metric learning with angular loss. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 2593–2601

[130] Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Proceedings of the International Conference on Neural Information Processing Systems. (2016) 1857–1865

[131] Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured

feature embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 4004–4012

[132] Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., Robertson, N.M.: Ranked list loss for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 5207–5216

[133] Chen, L., He, Y.: Dress fashionably: Learn fashion collocation with deep mixed-category metric learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2018) 2103–2110

[134] Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 360–368

[135] Kim, S., Kim, D., Cho, M., Kwak, S.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 3238–3247

[136] Zheng, W., Chen, Z., Lu, J., Zhou, J.: Hardness-aware deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 72–81

[137] Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2014) 1386–1393

[138] Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2015) 118–126

[139] Song, J., He, T., Gao, L., Xu, X., Shen, H.T.: Deep region hashing for efficient large-scale instance search from images. arXiv:1701.07901 (2017)

[140] Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision **124** (2017) 237–254

[141] Lin, J., Morere, O., Veillard, A., Duan, L.Y., Goh, H., Chandrasekhar, V.: Deephash for image instance retrieval: Getting regularization, depth and fine-tuning right. In: Proceedings of the ACM on International Conference on Multimedia Retrieval. (2017) 133–141

[142] Cao, J., Huang, Z., Wang, P., Li, C., Sun, X., Shen, H.T.: Quartet-net learning for visual instance retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2016) 456–460

[143] Wang, X., Zhang, H., Huang, W., Scott, M.R.: Cross-batch memory for embedding learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6388–6397

[144] Harwood, B., Kumar, B., Carneiro, G., Reid, I., Drummond, T., et al.: Smart mining for deep metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 2821–2829

[145] He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 596–605

[146] Revaud, J., Almazán, J., Rezende, R.S., Souza, C.R.d.: Learning with average precision: Training image retrieval with a listwise loss. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 5107–5116

[147] Brown, A., Xie, W., Kalogeiton, V., Zisserman, A.: Smooth-AP: Smoothing the path towards large-scale image retrieval. In: European Conference on Computer Vision. (2020) 677–694

[148] Aziere, N., Todorovic, S.: Ensemble deep manifold similarity learning using hard proxies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 7299–7307

[149] Donoser, M., Bischof, H.: Diffusion processes for retrieval revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2013) 1320–1327

[150] Iscen, A., Tolias, G., Avrithis, Y., Furon, T., Chum, O.: Efficient diffusion on region mani-

folds: Recovering small objects with compact CNN representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 2077–2086

[151] Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Mining on manifolds: Metric learning without labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 7642–7651

[152] Zhao, Y., Wang, L., Zhou, L., Shi, Y., Gao, Y.: Modelling diffusion process by deep neural networks for image retrieval. In: The British Machine Vision Conference. (2018) 161–174

[153] Song, B., Bai, X., Tian, Q., Latecki, L.J.: Regularized diffusion process on bidirectional context for object retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2018) 1213–1226

[154] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations. (2017)

[155] Maria, T., Anastasios, T.: Deep convolutional image retrieval: A general framework. Signal Processing: Image Communication **63** (2018) 30–43

[156] Tu, R.C., Mao, X.L., Feng, B.S., Yu, S.Y.: Object detection based deep unsupervised hashing. In: International Joint Conference on Artificial Intelligence. (2019) 3606–3612

[157] Zieba, M., Semberecki, P., El-Gaaly, T., Trzcinski, T.: BinGAN: learning compact binary descriptors with a regularized GAN. In: Proceedings of the International Conference on Neural Information Processing Systems. (2018) 3612–3622

[158] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2006) 2161–2168

[159] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2008) 1–8

[160] Radenovic, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018)

[161] Zheng, L., Wang, S., Wang, J., Tian, Q.: Accurate image search with multi-scale contextual evidences. International Journal of Computer Vision **120** (2016) 1–13

[162] Alzu'bi, A., Amira, A., Ramzan, N.: Content-based image retrieval with compact deep convolutional features. Neurocomputing **249** (2017) 95–105

[163] Valem, L.P., Pedronette, D.C.G.: Graph-based selective rank fusion for unsupervised image retrieval. Pattern Recognition Letters (2020)

[164] Alemu, L.T., Pelillo, M.: Multi-feature fusion for image retrieval using constrained dominant sets. Image and Vision Computing **94** (2020) 103862

[165] Yang, F., Hinami, R., Matsui, Y., Ly, S., Satoh, S.: Efficient image retrieval via decoupling diffusion into online and offline processing. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2019) 9087–9094

[166] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations. (2014)

[167] Husain, S.S., Bober, M.: REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval. IEEE Transactions on Image Processing **28** (2019) 5201–5213

[168] Iscen, A., Avrithis, Y., Tolias, G., Furon, T., Chum, O.: Fast spectral ranking for similarity search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 7632–7641

[169] Yang, J., Liang, J., Shen, H., Wang, K., Rosin, P.L., Yang, M.H.: Dynamic match kernel with deep convolutional features for image retrieval. IEEE Transactions on Image Processing **27** (2018) 5288–5302

[170] Yang, H.F., Lin, K., Chen, C.S.: Cross-batch reference learning for deep classification and

retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2016) 1237–1246

[171] Lv, Y., Zhou, W., Tian, Q., Sun, S., Li, H.: Retrieval oriented deep feature learning with complementary supervision mining. IEEE Transactions on Image Processing **27** (2018) 4945–4957

[172] Wang, Q., Lai, J., Claesen, L., Yang, Z., Lei, L., Liu, W.: A novel feature representation: Aggregating convolution kernels for image retrieval. Neural Networks **130** (2020) 1–10

[173] Jiang, Q.Y., Li, W.J.: Deep cross-modal hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3232–3240

[174] Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. (2013) 785–796

[175] Chi, J., Peng, Y.: Dual adversarial networks for zero-shot cross-media retrieval. In: International Joint Conference on Artificial Intelligence. (2018) 663–669

[176] Wang, K., Yin, Q., Wang, W., Wu, S., Wang, L.: A comprehensive survey on cross-modal retrieval. arXiv:1607.06215 (2016)

[177] Wang, B., Yang, Y., Xu, X., Hanjalic, A., Shen, H.T.: Adversarial cross-modal retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2017) 154–162

[178] Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 4242–4251

[179] Shannon, C.E.: A mathematical theory of communication. Bell system technical journal **27** (1948) 379–423

[180] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the International Conference on Neural Information Processing Systems. (2014) 2672–2680

[181] Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. arXiv:1409.7495 (2014)

[182] Huiskes, M.J., Lew, M.S.: The MIR Flickr retrieval evaluation. In: Proceedings of the ACM International Conference on Image and Video Retrieval. (2008) 39–43

[183] Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval. (2009) 48

[184] Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: Improving visual-semantic embeddings with hard negatives. In: The British Machine Vision Conference. (2018) 1–10

[185] Wang, D., Wang, Q., He, L., Gao, X., Tian, Y.: Joint and individual matrix factorization hashing for large-scale cross-modal retrieval. Pattern Recognition (2020) 107479

[186] Zhang, Y., Lu, H.: Deep cross-modal projection learning for image-text matching. In: European Conference on Computer Vision. (2018) 686–701

[187] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861 (2017)

[188] Graves, A., Fernández, S., Schmidhuber, J.: Bidirectional LSTM networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks. (2005) 799–804

[189] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. (2017) 1321–1330

[190] Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research **47** (2013) 853–899

[191] Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2** (2014) 67–78

[192] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision. (2014) 740–755

[193] Li, S., Xiao, T., Li, H., Zhou, B., Yue, D., Wang, X.: Person search with natural language description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 1970–1979

[194] Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 5005–5013

[195] Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9** (2008)

[196] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-RNN). In: International Conference on Learning Representations. (2015)

[197] Lev, G., Sadeh, G., Klein, B., Wolf, L.: RNN fisher vectors for action recognition and image annotation. In: European Conference on Computer Vision. (2016) 833–850

[198] Dong, J., Li, X., Snoek, C.G.: Predicting visual features from text for image and video caption retrieval. IEEE Transactions on Multimedia **20** (2018) 3377–3388

[199] Huang, Y., Wang, W., Wang, L.: Instance-aware image and sentence matching with selective multimodal LSTM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 2310–2318

[200] Liu, Y., Guo, Y., Bakker, E.M., Lew, M.S.: Learning a recurrent residual fusion network for multimodal matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 4127–4136

[201] Sarafianos, N., Xu, X., Kakadiaris, I.A.: Adversarial representation learning for text-to-image matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 5814–5824

[202] Nam, H., Ha, J.W., Kim, J.: Dual attention networks for multimodal reasoning and matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 299–307

[203] Zheng, Z., Zheng, L., Garrett, M., Yang, Y., Xu, M., Shen, Y.D.: Dual-path convolutional image-text embeddings with instance loss. ACM Transactions on Multimedia Computing, Communications, and Applications **16** (2020) 1–23

[204] Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 1890–1899

[205] Chen, D., Li, H., Liu, X., Shen, Y., Shao, J., Yuan, Z., Wang, X.: Improving deep visual representation for person re-identification by global and local image-language association. In: European Conference on Computer Vision. (2018) 54–70

[206] Niu, K., Huang, Y., Ouyang, W., Wang, L.: Improving description-based person re-identification by multi-granularity image-text alignments. IEEE Transactions on Image Processing **29** (2020) 5542–5556

[207] Klein, B., Lev, G., Sadeh, G., Wolf, L.: Associating neural word embeddings with deep image representations using fisher vectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2015) 4437–4446

[208] Bousquet, O., Elisseeff, A.: Stability and generalization. The Journal of Machine Learning Research **2** (2002) 499–526

[209] Yu, B.: Stability. Bernoulli **19** (2013) 1484–1500

[210] Sun, W.: Stability of machine learning algorithms. PhD thesis, Purdue University (2015)

[211] Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning. (2015) 1180–1189

[212] Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2017) 2935–2947

[213] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114** (2017) 3521–3526

[214] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)

[215] Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., Sriperumbudur, B.K.: Optimal kernel choice for large-scale two-sample tests. In: Proceedings of the International Conference on Neural Information Processing Systems. (2012) 1205–1213

[216] Zhai, M., Chen, L., Tung, F., He, J., Nawhal, M., Mori, G.: Lifelong GAN: Continual learning for conditional image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 2759–2768

[217] Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 3400–3409

[218] Wu, D., Dai, Q., Liu, J., Li, B., Wang, W.: Deep incremental hashing network for efficient image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 9069–9077

[219] Michieli, U., Zanuttigh, P.: Incremental learning techniques for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. (2019)

[220] Hou, S., Pan, X., Change Loy, C., Wang, Z., Lin, D.: Lifelong learning via progressive distillation and retrospection. In: European Conference on Computer Vision. (2018) 437–452

[221] Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Proceedings of the International Conference on Neural Information Processing Systems. (2017) 2990–2999

[222] Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Proceedings of the International Conference on Neural Information Processing Systems. (2016) 136–144

[223] Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford Dogs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop. (2011)

[224] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 dataset. (2011)

[225] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)

[226] Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 5022–5030

[227] Park, D., Hong, S., Han, B., Lee, K.M.: Continual learning by asymmetric loss approximation with single-side overestimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 3335–3344

[228] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. (2009)

[229] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern

Recognition. (2019) 374–382

[230] Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Proceedings of the International Conference on Neural Information Processing Systems. (2017) 6467–6476

[231] van de Ven, G.M., Tolias, A.S.: Generative replay with feedback connections as a general strategy for continual learning. arXiv:1809.10635 (2018)

[232] Yao, X., Huang, T., Wu, C., Zhang, R.X., Sun, L.: Adversarial feature alignment: Avoid catastrophic forgetting in incremental task lifelong learning. Neural Computation **31** (2019) 2266–2291

[233] Parshotam, K., Kilickaya, M.: Continual learning of object instances. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2020) 224–225

[234] Xie, L., Wang, J., Zhang, B., Tian, Q.: Fine-grained image search. IEEE Transactions on Multimedia **17** (2015) 636–647

[235] Zhou, P., Mai, L., Zhang, J., Xu, N., Wu, Z., Davis, L.S.: M2KD: Multi-model and multi-level knowledge distillation for incremental learning. In: The British Machine Vision Conference. (2020) 1–10

[236] Tian, X., Ng, W., Wang, H., Kwong, S.: Complementary incremental hashing with query-adaptive re-ranking for image retrieval. IEEE Transactions on Multimedia (2020) 1–15

[237] Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision (2021)

[238] Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 4133–4141

[239] Huang, X., Peng, Y.: TPCKT: two-level progressive cross-media knowledge transfer. IEEE Transactions on Multimedia **21** (2019) 2850–2862

[240] Ma, X., Zhang, T., Xu, C.: Multi-level correlation adversarial hashing for cross-modal retrieval. IEEE Transactions on Multimedia **22** (2020) 3101–3114

[241] Peng, Y., Qi, J.: Show and tell in the loop: Cross-modal circular correlation learning. IEEE Transactions on Multimedia **21** (2018) 1538–1550

[242] Peng, Y., Qi, J., Huang, X., Yuan, Y.: CCL: Cross-modal correlation learning with multi-grained fusion by hierarchical network. IEEE Transactions on Multimedia **20** (2017) 405–420

[243] Li, Z., Tang, J., Mei, T.: Deep collaborative embedding for social image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2018) 2070–2083

[244] Peng, Z., Li, Z., Zhang, J., Li, Y., Qi, G.J., Tang, J.: Few-shot image recognition with knowledge transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 441–449

[245] Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 1365–1374

[246] Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: European Conference on Computer Vision. (2018) 532–547

[247] Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep feature interpolation for image content changes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 7064–7073

[248] Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2010) 117–128

[249] Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 3366–3375

[250] Perez-Rua, J.M., Zhu, X., Hospedales, T.M., Xiang, T.: Incremental few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 13846–13855

[251] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113** (2019) 54–71

[252] French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3** (1999) 128–135

[253] Yu, L., Yazici, V.O., Liu, X., Weijer, J.v.d., Cheng, Y., Ramisa, A.: Learning metrics from teachers: Compact networks for image embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2907–2916

[254] Lu, J., Hu, J., Zhou, J.: Deep metric learning for visual understanding: An overview of recent advances. IEEE Signal Processing Magazine **34** (2017) 76–84

[255] Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 8715–8724

[256] Haroush, M., Hubara, I., Hoffer, E., Soudry, D.: The knowledge within: Methods for data-free model compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 8494–8502

[257] Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 5007–5016

[258] Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 3967–3976

[259] Rannen, A., Aljundi, R., Blaschko, M.B., Tuytelaars, T.: Encoder based lifelong learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 1320–1328

[260] Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. (2013) 554–561

[261] Wei, K., Deng, C., Yang, X.: Lifelong zero-shot learning. In: International Joint Conference on Artificial Intelligence. (2020) 551–557

[262] Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Proceedings of the International Conference on Neural Information Processing Systems. (2016)

[263] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the International Conference on Neural Information Processing Systems. (2017)

[264] Park, C.C., Kim, Y., Kim, G.: Retrieval of sentence sequences for an image stream via coherence recurrent convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2018) 945–957

[265] Liang, J., Jiang, L., Cao, L., Kalantidis, Y., Li, L.J., Hauptmann, A.G.: Focal visual-text attention for memex question answering. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2019) 1893–1908

[266] Chen, H., Ding, G., Lin, Z., Zhao, S., Han, J.: Show, observe and tell: Attribute-driven attention model for image captioning. In: International Joint Conference on Artificial Intelligence. (2018) 606–612

[267] Cha, M., Gwon, Y.L., Kung, H.: Adversarial learning of semantic relevance in text to image synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 3272–3279

[268] Gorti, S.K., Ma, J.: Text-to-image-to-text translation using cycle consistent adversarial networks. arXiv:1808.04538 (2018)

[269] Wu, L., Wang, Y., Shao, L.: Cycle-consistent deep generative hashing for cross-modal retrieval. IEEE Transactions on Image Processing **28** (2018) 1602–1612

[270] Yao, T., Pan, Y., Li, Y., Mei, T.: Exploring visual relationship for image captioning. In: European Conference on Computer Vision. (2018) 684–699

[271] Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 10685–10694

[272] Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 7008–7024

[273] Liu, D., Zha, Z.J., Zhang, H., Zhang, Y., Wu, F.: Context-aware visual policy network for sequence-level image captioning. arXiv:1808.05864 (2018)

[274] Hossain, M., Sohel, F., Shiratuddin, M.F., Laga, H.: A comprehensive survey of deep learning for image captioning. ACM Computing Surveys (CSUR) **51** (2019) 118

[275] Wang, H., Wang, H., Xu, K.: Evolutionary recurrent neural network for image captioning. Neurocomputing (2020)

[276] Dai, B., Fidler, S., Urtasun, R., Lin, D.: Towards diverse and natural image descriptions via a conditional GAN. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 2970–2979

[277] Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Proceedings of the International Conference on Neural Information Processing Systems. (2015) 2440–2448

[278] Park, C.C., Kim, B., Kim, G.: Towards personalized image captioning via multimodal memory networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)

[279] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 6077–6086

[280] Song, L., Liu, J., Qian, B., Chen, Y.: Connecting language to images: A progressive attention-guided network for simultaneous image captioning and language grounding. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8885–8892

[281] Jin, J., Nakayama, H.: Annotation order matters: Recurrent image annotator for arbitrary length image tagging. In: International Conference on Pattern Recognition. (2016) 2452–2457

[282] Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 1261–1270

[283] Anderson, P., Gould, S., Johnson, M.: Partially-supervised image captioning. In: Proceedings of the International Conference on Neural Information Processing Systems. (2018) 1879–1890

[284] El, O.B., Licht, O., Yosephian, N.: GILT: Generating images from long text. arXiv:1901.02404 (2019)

[285] Mansimov, E., Parisotto, E., Ba, J.L., Salakhutdinov, R.: Generating images from captions with attention. International Conference on Learning Representations (2016)

[286] Reed, S., van den Oord, A., Kalchbrenner, N., Colmenarejo, S.G., Wang, Z., Chen, Y., Belov, D., de Freitas, N.: Parallel multiscale autoregressive density estimation. In: International Conference on Machine Learning. (2017) 2912–2921

[287] Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 1219–1228

[288] Hong, S., Yang, D., Choi, J., Lee, H.: Inferring semantic layout for hierarchical text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 7986–7994

[289] Zhang, Z., Xie, Y., Yang, L.: Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 6199–6208

[290] Gao, L., Chen, D., Song, J., Xu, X., Zhang, D., Shen, H.T.: Perceptual pyramid adversarial networks for text-to-image synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2019) 8312–8319

[291] Han, Z., Tao, X., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN++: Realistic image synthesis with stacked generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018) 1–1

[292] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 1316–1324

[293] Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: Proceedings of the International Conference on Neural Information Processing Systems. (2016) 217–225

[294] Yuan, M., Peng, Y.: Text-to-image synthesis via symmetrical distillation networks. In: Proceedings of the ACM International Conference on Multimedia. (2018) 1407–1415

[295] Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 49–58

[296] Zhang, S., Dong, H., Hu, W., Guo, Y., Wu, C., Xie, D., Wu, F.: Text-to-image synthesis via visual-memory creative adversarial network. In: Pacific Rim Multimedia. (2018) 417–427

[297] Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Proceedings of the International Conference on Neural Information Processing Systems. (2017) 3856–3866

[298] Song, G., Wang, D., Tan, X.: Deep memory network for cross-modal retrieval. IEEE Transactions on Multimedia **21** (2018) 1261–1275

[299] Zhang, M., Zhang, H., Li, J., Wang, L., Fang, Y., Sun, J.: Supervised graph regularization based cross media retrieval with intra and inter-class correlation. Journal of Visual Communication and Image Representation **58** (2019) 1–11

[300] Wu, Y., Wang, S., Song, G., Huang, Q.: Augmented adversarial training for cross-modal retrieval. IEEE Transactions on Multimedia (2020)

[301] Wang, Y., Luo, X., Nie, L., Song, J., Zhang, W., Xu, X.S.: BATCH: A scalable asymmetric discrete cross-modal hashing. IEEE Transactions on Knowledge and Data Engineering (2020)

[302] Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., van den Hengel, A.: Visual question answering: A survey of methods and datasets. Computer Vision and Image Understanding **163** (2017) 21–40

[303] Teney, D., Anderson, P., He, X., van den Hengel, A.: Tips and tricks for visual question answering: Learnings from the 2017 challenge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 4223–4232

[304] Li, Q., Tao, Q., Joty, S., Cai, J., Luo, J.: VQA-E: Explaining, elaborating, and enhancing your answers for visual questions. In: European Conference on Computer Vision. (2018) 552–567

[305] Zhang, Y., Hare, J.S., Prügel-Bennett, A.: Learning to count objects in natural images for visual question answering. International Conference on Learning Representations (2018)

[306] Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., Parikh, D.: Yin and yang: Balancing and answering binary visual questions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2016) 5014–5022

[307] Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Computational intelligence magazine **13** (2018) 55–75

[308] Liu, X., Wang, M., Zha, Z.J., Hong, R.: Cross-modality feature learning via convolutional

autoencoder. ACM Transactions on Multimedia Computing, Communications, and Applications **15** (2019) 7

[309] Xu, X., Song, J., Lu, H., Yang, Y., Shen, F., Huang, Z.: Modal-adversarial semantic learning network for extendable cross-modal retrieval. In: Proceedings of the ACM on International Conference on Multimedia Retrieval. (2018) 46–54

[310] Zhu, X., Li, L., Liu, J., Li, Z., Peng, H., Niu, X.: Image captioning with triple-attention and stack parallel LSTM. Neurocomputing **319** (2018) 55–65

[311] Jiang, W., Ma, L., Jiang, Y.G., Liu, W., Zhang, T.: Recurrent fusion network for image captioning. In: European Conference on Computer Vision. (2018) 499–515

[312] Chen, C., Mu, S., Xiao, W., Ye, Z., Wu, L., Ju, Q.: Improving image captioning with conditional generative adversarial nets. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8142–8150

[313] Dash, A., Gamboa, J.C.B., Ahmed, S., Liwicki, M., Afzal, M.Z.: TAC-GAN-text conditioned auxiliary classifier generative adversarial network. arXiv:1703.06412 (2017)

[314] Gu, J., Joty, S., Cai, J., Zhao, H., Yang, X., Wang, G.: Unpaired image captioning via scene graph alignments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 10323–10332

[315] Xu, W., Keshmiri, S., Wang, G.R.: Adversarially approximated autoencoder for image generation and manipulation. IEEE Transactions on Multimedia (2019)

[316] Feng, Y., Ma, L., Liu, W., Luo, J.: Unsupervised image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 4125–4134

[317] Yu, J., Yang, C., Qin, Z., Yang, Z., Hu, Y., Liu, Y.: Textual relationship modeling for cross-modal information retrieval. arXiv (2018)

[318] Chen, F., Ji, R., Su, J., Wu, Y., Wu, Y.: Structcap: Structured semantic embedding for image captioning. In: Proceedings of the ACM International Conference on Multimedia. (2017) 46–54

[319] Teney, D., Liu, L., van den Hengel, A.: Graph-structured representations for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 1–9

[320] Zhang, D., Cao, R., Wu, S.: Information fusion in visual question answering: A survey. Information Fusion (2019)

[321] Su, Z., Zhu, C., Dong, Y., Cai, D., Chen, Y., Li, J.: Learning visual knowledge memory networks for visual question answering. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 7736–7745

[322] Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. arXiv:1803.03067 (2018)

[323] Fan, H., Zhou, J.: Stacked latent attention for multimodal reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 1072–1080

[324] Xiong, C., Merity, S., Socher, R.: Dynamic memory networks for visual and textual question answering. In: International Conference on Machine Learning. (2016) 2397–2406

[325] Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: European Conference on Computer Vision. (2016) 451–466

[326] Ma, C., Shen, C., Dick, A.R., Wu, Q., Wang, P., van den Hengel, A., Reid, I.D.: Visual question answering with memory-augmented networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 6975–6984

[327] Wu, C., Liu, J., Wang, X., Dong, X.: Object-difference attention: A simple relational attention for visual question answering. In: Proceedings of the ACM International Conference on Multimedia. (2018) 519–527

[328] Singh, J., Ying, V., Nutkiewicz, A.: Attention on attention: Architectures for visual question answering (VQA). arXiv:1803.07724 (2018)

[329] Nguyen, D.K., Okatani, T.: Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 6087–6096

[330] Agrawal, A., Batra, D., Parikh, D., Kembhavi, A.: Don't just assume; look and answer: Overcoming priors for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 4971–4980

[331] Yu, Z., Yu, J., Fan, J., Tao, D.: Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 1821–1830

[332] Wang, W., Liu, P., Yang, S., Zhang, W.: Dynamic interaction networks for image-text multimodal learning. Neurocomputing **379** (2020) 262–272

[333] Peng, G., Li, H., You, H., Jiang, Z., Lu, P., Hoi, S., Wang, X.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. arXiv:1812.05252 (2018)

[334] Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2019) 423–443

[335] Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2016) 457–468

[336] Kim, J.H., On, K.W., Lim, W., Kim, J., Ha, J.W., Zhang, B.T.: Hadamard product for low-rank bilinear pooling. arXiv:1610.04325 (2016)

[337] Ben-Younes, H., Cadene, R., Cord, M., Thome, N.: MUTAN: Multimodal tucker fusion for visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 2612–2620

[338] Gao, P., Li, H., Li, S., Lu, P., Li, Y., Hoi, S.C., Wang, X.: Question-guided hybrid convolution for visual question answering. In: European Conference on Computer Vision. (2018) 469–485

[339] Liu, X., Li, H., Shao, J., Chen, D., Wang, X.: Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In: European Conference on Computer Vision. (2018) 338–354

[340] Wu, Q., Wang, P., Shen, C., Reid, I., van den Hengel, A.: Are you talking to me? reasoned visual dialog generation through adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 6106–6115

[341] Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8** (1992) 229–256

[342] Gao, J., Wang, S., Wang, S., Ma, S., Gao, W.: Self-critical n-step training for image captioning. arXiv:1904.06861 (2019)

[343] Li, Y., Duan, N., Zhou, B., Chu, X., Ouyang, W., Wang, X., Zhou, M.: Visual question generation as dual task of visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 6116–6124

[344] Li, C., Deng, C., Wang, L., Xie, D., Liu, X.: Coupled CycleGAN: Unsupervised hashing network for cross-modal retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2019) 176–183

[345] Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2015) 2641–2649

[346] You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 4651–4659

[347] Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via

a visual sentinel for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 375–383

[348] Yan, S., Wu, F., Smith, J.S., Lu, W., Zhang, B.: Image captioning using adversarial networks and reinforcement learning. In: International Conference on Pattern Recognition. (2018) 248–253

[349] Liu, F., Xiang, T., Hospedales, T.M., Yang, W., Sun, C.: Inverse visual question answering: A new benchmark and vqa diagnosis tool. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)

[350] Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 12174–12182

[351] Lao, Q., Havaei, M., Pesaranghader, A., Dutil, F., Jorio, L.D., Fevens, T.: Dual adversarial inference for text-to-image synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2019) 7567–7576

[352] Joseph, K., Pal, A., Rajanala, S., Balasubramanian, V.N.: C4Synth: Cross-caption cycle-consistent text-to-image synthesis. In: IEEE Winter Conference on Applications of Computer Vision. (2019) 358–366

[353] Qiao, T., Zhang, J., Xu, D., Tao, D.: MirrorGAN: Learning text-to-image generation by re-description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 1505–1514

[354] Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.: Controllable text-to-image generation. arXiv:1909.07083 (2019)

[355] Yin, G., Liu, B., Sheng, L., Yu, N., Wang, X., Shao, J.: Semantics disentangling for text-to-image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 2327–2336

[356] Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Parikh, D., Batra, D.: VQA: Visual question answering. International Journal of Computer Vision **123** (2017) 4–31

[357] Schwartz, I., Schwing, A., Hazan, T.: High-order attention models for visual question answering. In: Proceedings of the International Conference on Neural Information Processing Systems. (2017) 3664–3674

[358] Yu, D., Fu, J., Mei, T., Rui, Y.: Multi-level attention networks for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 4709–4717

[359] Zhu, C., Zhao, Y., Huang, S., Tu, K., Ma, Y.: Structured attentions for visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2017) 1291–1300

[360] Lu, P., Li, H., Zhang, W., Wang, J., Wang, X.: Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2018)

[361] Song, J., Zeng, P., Gao, L., Shen, H.T.: From pixels to objects: Cubic visual attention for visual question answering. In: International Joint Conference on Artificial Intelligence. (2018) 906–912

[362] Osman, A., Samek, W.: Dual recurrent attention units for visual question answering. arXiv:1802.00209 (2018)

[363] Liu, Y., Zhang, X., Huang, F., Li, Z.: Adversarial learning of answer-related representation for visual question answering. In: Proceedings of the ACM International Conference on Information and Knowledge Management. (2018) 1013–1022

[364] Wu, C., Liu, J., Wang, X., Li, R.: Differential networks for visual question answering. Proceedings of the AAAI Conference on Artificial Intelligence (2019)

[365] Liu, F., Liu, J., Fang, Z., Lu, H.: Language and visual relations encoding for visual question answering. In: IEEE International Conference on Image Processing. (2019) 3307–3311

[366] Liu, F., Liu, J., Fang, Z., Hong, R., Lu, H.: Densely connected attention flow for visual question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2019) 869–875

[367] Aytar, Y., Vondrick, C., Torralba, A.: See, hear, and read: Deep aligned representations. arXiv:1706.00932 (2017)

[368] He, X., Peng, Y., Xie, L.: A new benchmark and approach for fine-grained cross-media retrieval. In: Proceedings of the ACM International Conference on Multimedia. (2019) 1740–1748

[369] Fu, C., Pei, W., Cao, Q., Zhang, C., Zhao, Y., Shen, X., Tai, Y.W.: Non-local recurrent neural memory for supervised sequence modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 6311–6320

List of Abbreviations

Abbreviation	Full Name / Short Definition
DCNNs	Deep Convolutional Neural Networks / A regularized version of multilayer perceptrons based on convolution kernels
CBIR	Content-based Image Retrieval / An image search task according to the content contained in images
MAC	Maximum Activations of Convolutions / Maximum value over a convolutional feature map
R-MAC	Regional Maximum Activations of Convolutions / Maximum value over a region on a convolutional feature map
CroW	Cross-dimensional Weighting / Weighting the activations over different feature maps
SPoC	Sum-Pooled Convolutional / Sum pooling over different feature maps
ReLU	Rectified Linear Unit / An activation function returns 0 if it receives any negative input
SPM	Spatial Pyramid Modeling / An method to model feature in a pyramid way
t-SNE	t-Distributed Stochastic Neighbor Embedding / A method to visualize high-dimensional data
RPNs	Region Proposal Networks / A network to obtain proposal for a region or an object
FC	Fully-Connected (layer)
KNN	K-Nearest Neighbors
BoW	Bag-of-Words / Method to embed features according to the number of feature occurrences
VLAD	Vector of Locally Aggregated Descriptors / Method to embed features based on their residuals w.r.t. each visual word
FV	Fisher Vector / Method to embed features by using Gaussian mixture model
GeM	Generalized Mean / A pooling method to apply over each feature map
CAM	Class Activation Maps / A feature weighting method based on an activated class output
PCA	Principal Component Analysis
MMD	Maximum Mean Discrepancy
FGIR	Fine-Grained Image Retrieval
RKHS	Reproducing Kernel Hilbert Space
DKD	Dual Knowledge Distillation // Knowledge distillation based on two teacher models
GCNs	Graph Convolutional Networks
VQA	Visual Question Answering / A computer vision task
KL-divergence	Kullback–Leibler divergence / A metric to measure the distance between two distributions

English Summary

We are living in an information era where the amount of image and video data increases exponentially. It is important to develop appropriate information systems to store, manage, and distribute such large data collections. Among them, intelligent image retrieval is one of the most indispensable techniques to be considered. It satisfies our needs for searching information of interest. To enable intelligent image retrieval (including high accuracy and high efficiency retrieval), feature representations are at the core of most retrieval algorithms.

For humans, it is easy to find similar images from an image gallery according to a given query image. However, it is difficult for a computer to search as accurately as humans due to the existing semantic gap between the high-level concepts used by humans and the typically low-level features derived from images (*i.e.* pixels or symbols). In addition, it will be more difficult for the computer to search accurately if the query contains multiple modalities (*e.g.* text, audio *etc*). This is caused by the second challenge: the heterogeneity gap. Deep learning, especially for convolutional neural networks has made progress in addressing these challenges and significantly facilitated the process of intelligent image retrieval.

The first theme in this thesis is to explore cross-modal retrieval by considering visual and textual modalities. This theme is hard to realize because it involves both the above mentioned semantic gap and heterogeneity gap. We design an information entropy loss function based on Shannon information theory to regularize the learning of a shared latent space for paired image and text inputs. The common practice of cross-modal retrieval is to construct a shared space where image features and text features are highly intermixed, thereby the similarity between image and text can be further associated. This property of the shared space is consistent with Shannon information theory by measuring the information entropy. This idea is demonstrated for cross-modal hashing retrieval where real-valued features and binary hash codes are constrained by the information entropy loss.

Next, we explore the integration of Shannon information theory and adversarial learning for cross-modal retrieval. This adversarial mechanism achieves a better feature distribution agreement for the two modalities thereby bridging the heterogeneity gap and enabling a more accurate retrieval. To reduce the semantic gap, Kullback-Leibler (KL) divergence and bi-directional triplet loss are used to associate

the intra- and inter-modality similarity between features in the shared space. Also, we design a regularization term based on KL-divergence with temperature scaling to calibrate the bias of the label classifier that is caused by the data imbalance issue.

The second theme of this thesis is to explore the continuous retrieval capacity of deep neural networks where three important sub-questions are studied: incremental learning for retrieval on the same fine-grained dataset, feature estimation for sequences of deep models in incremental learning, and lifelong learning for image retrieval on different datasets, respectively. Unlike the learning process of humans, training previously trained deep networks on new data leads the networks to forget what was learned before. For the first sub-question, we employ incremental learning for the find-grained image retrieval task. This is achieved through regularizing the retrieval representations and classification probabilities by using a maximum mean discrepancy loss function and knowledge distillation loss function. To evaluate the proposed method, we split a dataset into two parts, one is used as the old data (or old tasks) and the other is used as the new data for incremental training (or new tasks).

For the second sub-question, we focus on the sequence of deep neural networks which have been trained when new tasks are added sequentially. This multi-task scenario will suffer from more severe catastrophic forgetting. Saving the sequence of models for transferring previously learned knowledge is memory-consuming. Instead, we propose a simple but effective feature estimation method to alleviate this limitation.

For the third sub-question, we consider a more practical lifelong image retrieval scenario where the deep model is trained successively on different datasets. The semantic drifts between different datasets make minimizing the forgetting ratio more difficult. We address this limitation by using a dual knowledge distillation framework that includes two professional teachers and a self-motivated student. One teacher model has its parameters fixed and is used for transferring previously learned knowledge on the proceeding tasks while another on-the-fly teacher is trained jointly with the student and is responsible for transferring knowledge learned on the newly added tasks. Furthermore, we also use the statistics on the BatchNorm layers of the frozen teacher model to generate some representative images for the successive training tasks.

We conduct thorough experiments to verify the efficacy of the proposed methods for the two themes. The results demonstrate significant improvements over various baselines and state-of-the-art methods. Therefore, this thesis provides novel contributions, insights, and findings for the research community and future applications in the field of intelligent image retrieval.

Nederlandse Samenvatting

We leven in een informatietijdperk. De hoeveelheid beeld- en video-gegevens neemt exponentieel toe. Het is belangrijk om informatiesystemen te ontwikkelen die in staat zijn om zulke grote gegevenscollecties op te slaan, te beheren en te verspreiden. Een van de belangrijkste technieken hierbij is het intelligent ophalen van afbeeldingen; dit is een van de meest onmisbare technieken om te voldoen aan onze behoefte om visuele informatie die van belang is te vinden. Om het mogelijk te maken afbeeldingen intelligent op te halen zijn verschillende algoritmen noodzakelijk, hierbij moet gelet worden op hoge nauwkeurigheid en hoge efficiëntie. Representatiefuncties vormen hierbij de kern van verschillende ophaalalgoritmen die we in dit proefschrift zullen bespreken.

Voor mensen is het eenvoudig om vergelijkbare afbeeldingen uit een verzameling beelden te vinden op basis van een gegeven voorbeeld. Het is echter moeilijk voor een computer om even nauwkeurig te zoeken als mensen vanwege de bestaande semantische kloof tussen de concepten gebruikt door mensen en de beeldkenmerken die gewoonlijk worden afgeleid van afbeeldingen (dwz pixels of symbolen). Bovendien zal het voor de computer moeilijker zijn om nauwkeurig te zoeken als het zoekitem uit verschillende modaliteiten bestaat (*e.g.* tekst, audio *etc*). Dit wordt veroorzaakt door de tweede uitdaging: de heterogeniteitskloof. Deep learning heeft, vooral voor convolutionele neurale netwerken, vooruitgang geboekt bij het aanpakken van deze uitdagingen en het proces van het intelligent ophalen van afbeeldingen aanzienlijk vergemakkelijkt.

Het eerste onderwerp in dit proefschrift is het verkennen van multi-modale retrieval door zowel de visuele als de tekstuele modaliteiten te gebruiken. De moeilijkheid bij dit onderwerp zit hem in de semantische kloof en de heterogeniteitskloof. We ontwerpen een informatie-entropieverliesfunctie op basis van Shannon's informatietheorie om het leren van een gedeelde latente ruimte voor gepaarde beeld- en tekstinvoer te regulariseren. De gebruikelijke praktijk van cross-modale opvraging is om een gedeelde ruimte te construeren waar afbeeldingskenmerken en tekstkenmerken sterk door elkaar worden gehaald, waardoor de overeenkomst tussen afbeelding en tekst verder kan worden geassocieerd. Deze eigenschap van de gedeelde ruimte is consistent met de meting van informatie-entropie volgens Shannon's informatietheorie. Dit idee wordt gedemonstreerd voor cross-modale hashing retrieval waarbij de reële

featurewaarden en de binaire hashcodes worden beperkt/gestuurd door het verlies van informatie-entropie.

Vervolgens integreren we Shannon's informatietheorie en adversarial learning voor cross-modale retrieval. Adversarial learning zorgt voor een betere verdeling van bimodale-kernmerken opdat we de heterogeniteitskloof kunnen overbruggen en zo betere prestaties mogelijk maken. Om de semantische kloof te verkleinen, worden Kullback-Leibler (KL) divergentie en bidirectioneel tripletverlies gebruikt om de intra- en inter-modaliteitsgelijkvormigheid tussen kenmerken in de gemeenschappelijke ruimte te vinden. We ontwerpen ook een regularisatieterm op basis van KL-divergentie met temperatuurschaling om de bevooroordeelde labelclassificatie te kalibreren en zo de onbalans in de basisdata te verminderen.

Het tweede thema van dit proefschrift betreft de vraag hoe we leer-taken van een voorgaande taak kunnen leren, zonder telkens overnieuw te moeten beginnen, of het geleerde te vergeten als nieuwe zaken geleerd worden. We onderscheiden drie belangrijke deelvragen: incrementeel leren voor retrieval in dezelfde fijnmazige dataset, feature-schattingen voor opeenvolgende diepe modellen in incrementeel leren en levenslang leren voor rerieval in verschillende datasets. Voor de eerste deelvraag kijken we naar incrementeel leren voor het vinden van fine-grained afbeeldingen. Dit wordt bereikt door de representatie- en classificatie-distributies te regulariseren. Dit doen we door gebruik te maken van het maximale gemiddelde discrepantieverlies en kennisdestillatieverlies. Om de voorgestelde methode te evalueren, splitsen we een dataset in twee delen, de ene wordt gebruikt als oude data (of oude taken) en de andere wordt gebruikt als de nieuwe data voor incrementele training (of nieuwe taken).

Voor de tweede deelvraag richten we ons op de sequentie van diepe modellen die worden getraind wanneer nieuwe taken opeenvolgend worden toegevoegd. Dit scenario met meerdere taken zal lijden aan catastrofaal vergeten. Het opslaan van de sequentie van modellen voor het overdragen van eerder geleerde kennis is geheugenverslindend. In plaats daarvan stellen we een eenvoudige maar effectieve methode voor het schatten van features voor om deze beperking te verminderen.

Voor de derde deelvraag kijken we naar de praktische kant van levenslang leren voor het zoeken naar beelden, waarbij het neurale network achtereenvolgens wordt getraind op verschillende datasets. De semantische verschuivingen tussen verschillende datasets maken het moeilijk om iets te doen aan het vergeten van geleerde features. We pakken deze beperking aan door een duaal kennisdestillatiekader te gebruiken dat twee professionele supervisors en een instrinsiek-gemotiveerde student omvat. Het ene supervisormodel heeft vaste parameters en wordt gebruikt voor het overdragen van eerder geleerde kennis aan de volgende taken, terwijl een andere on-the-fly supervisor samen met de student wordt opgeleid en verantwoordelijk is voor het overdragen van kennis die is geleerd over de nieuw toegevoegde taken. Verder

gebruiken we ook de statistieken over de BatchNorm-lagen van het bevroren supervisormodel om enkele representatieve afbeeldingen te genereren voor de volgende taken.

Tenslotte voeren we diepgaande experimenten uit om de effectiviteit van de voorgestelde methoden voor de twee onderwerpen vast te stellen. De resultaten laten significante verbeteringen zien ten opzichte van verschillende baselines en state-of-the-art methoden. Dit proefschrift levert nieuwe bijdragen, inzichten en vondsten voor de onderzoeksgemeenschap en toekomstige toepassingen op het gebied van intelligente image-retrieval.

Acknowledgements

This thesis is a reward for my academic journey, which is challenging, encouraging, and memorable. Indeed, my PhD journey would not have been possible without the support and guidance from my supervisors, and encouragement from family, colleagues, and friends.

First of all, I would like to express my gratitude to my supervisors, Professor Michael Lew and Professor Aske Plaat, who gave me the opportunity to study at the MediaLab. It is my great honor to work with you throughout my doctoral studies. What I have learned is not only the specialized knowledge in the field of computer vision and multimedia but also the spirit of being an independent academic researcher. Thank you for your supervision and insightful suggestions during my doctoral studies.

I would like to deliver my appreciation to Dr. Erwin Bakker. Erwin gave me a lot of inspiration for each of my presentations. Also, he gave me many constructive suggestions and comments for each of my paper submissions.

I would also like to thank Hyowon Kim for numerous helpful suggestions for phrasing and editing my papers.

I would like to express my appreciation to my doctorate committee members: Professor Tat-Seng Chua, Professor Boudewijn P.F. Lelieveldt, Professor Thomas Bäck, Dr. Erwin Bakker, and Dr. Katy Wolstencroft. They reviewed my thesis carefully and gave me insightful comments and suggestions.

My special appreciation goes to my Chinese supervisor Professor Weiping Wang. I am very grateful for his continuous support for my Master study. I would like to express my gratitude to Professor Li Liu, for your encouragement and support, helping me to overcome the difficulties in my research.

I would like to express my gratitude to my colleagues at the MediaLab: Yanming Guo, Yu Liu, Theodoros Georgiou, Nan Pu, and Mingrui Lao. Yanming Guo helped me a lot when I first arrived at Leiden University. I was also inspired by the discussions with Yu Liu, Nan Pu, and Mingrui Lao. Thank you for sharing your knowledge. I also enjoyed and benefited from the discussions and suggestions with Theodoros Georgiou, Nan Pu, and Mingrui Lao.

I would like to express my overwhelming gratitude to Zhihan Zhao (Shui Bao). During 1440 days-and so much more, we accompany, encourage, cheer each other. To be a better you is to be a better me. We feel happy, angry, moved, frustrated. At this moment, every second is memorable. You fill my world with light and with joy. Most importantly, I believe in destiny: from the Hubrecht Institute to Leiden University Medical Center, from Utrecht to Leiden. I cherish and am grateful!

I would like to express my deepest appreciation to my parents for their unconditional love and support. When I became depressed during my doctoral research, I was always encouraged. When I tried something new, I was always supported. I learned perseverance and persistence from them. I hope they are happy and healthy. Moreover, I share my appreciation to my older brother, sister-in-law, and younger sister.

Last but not least, I feel lucky to have made so many Chinese friends in Leiden and in the Netherlands. The days with them made my doctoral studies more enjoyable and less stressful.

Wei Chen
July 2021
Leiden, the Netherlands

Curriculum Vitae

Wei Chen was born in Guizhou, China on July 16, 1991. In 2010, he started his bachelor's study at Wuhan University in Wuhan, Hubei, China, and received his bachelor's degree in 2014. After that, he started his master's study at National University of Defense Technology in Changsha, Hunan, China, and obtained his master's degree in 2016 under the supervision of Prof. Dr. Weiping Wang.

In September 2017, he started his PhD research supported by the China Scholarship Council (CSC No. 201703170183) and worked at the MediaLab in the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands, under the supervision of Prof. Dr. Michael Lew and Prof. Dr. Aske Plaat. Wei Chen's research interests include computer vision, multimedia retrieval and deep learning. He is focusing on image retrieval in the context of incremental learning. He has published papers in international journals and conferences, including IEEE Transactions on Multimedia, Pattern Recognition, Neurocomputing, CVPR, ACM MM, ICME, BMVC, *etc.* He also serves as a reviewer for some conferences and journals, such as IEEE Transactions on Multimedia, Pattern Recognition, Neurocomputing, ICME, and BMVC.