



Universiteit
Leiden
The Netherlands

Exploring images with deep learning for classification, retrieval and synthesis

Liu, Y.

Citation

Liu, Y. (2018, October 24). *Exploring images with deep learning for classification, retrieval and synthesis*. *ASCI dissertation series*. Retrieved from <https://hdl.handle.net/1887/66480>

Version: Not Applicable (or Unknown)

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/66480>

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

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/66480> holds various files of this Leiden University dissertation.

Author: Liu, Y.

Title: Exploring images with deep learning for classification, retrieval and synthesis

Issue Date: 2018-10-24

Chapter 7

Joint Matching and Classification

In Chapters 2-6, we have proposed several methods to solve the classification and retrieval themes, separately. Unlike many existing approaches which focus only on either multi-modal matching or classification, we aim to study how we can integrate the two tasks together to help promote each other (RQ6).

In this chapter, we propose a unified **Network** to jointly learn **Multi-modal Matching** and **Classification** (MMC-Net) between images and texts. The proposed MMC-Net model can seamlessly integrate the matching and classification components. It first learns visual and textual embedding features in the matching component, and then generates discriminative multi-modal representations in the classification component. Combining the two components in a unified model can help in improving their performance simultaneously. Moreover, we present a multi-stage training algorithm by minimizing both of the matching and classification loss functions. Experimental results on four well-known multi-modal benchmarks demonstrate the effectiveness and efficiency of the proposed approach, which achieves competitive performance for multi-modal matching and classification compared to the state-of-the-art approaches.

Keywords

Multi-modal matching, Multi-modal classification, Deep neural networks, Multi-stage training

7.1 Introduction

The problem of multi-modal analytic has attracted increasing attention due to a drastic growth of multimedia data such as image, video and text. Particularly, multi-modal matching has been studied for decades, with the aim of searching for a latent space, where visual and textual features can be unified to be latent embeddings. The hypothesis is that different modalities have semantically related properties that can be distilled into a common latent space. Early approaches to learning latent embeddings are based on the Canonical Correlation Analysis (CCA) [61], which is effective at maximizing the high correlation between visual and textual features in the latent space. Driven by the increasing progress of deep learning, many works [52, 55, 66, 181] have been dedicated to developing deep matching networks to learn discriminative latent embeddings and train the networks by using a bi-directional rank loss function. They have achieved state-of-the-art performance on many well-known multi-modal benchmarks [53, 64, 67, 76].

However, learning latent embeddings is influenced by the notable variance in images or texts. For example, in Figure 7.1, five sentences annotated by humans are provided to describe the same image. The input image and five sentences are projected into a latent space. One can observe that these sentences have significant variance on representing the visual content. Although they can consistently describe the main objects in the scene including ‘girl’ (or ‘child’) and ‘bicycle’ (or ‘bike’), they still present great variance in terms of other objects, *e.g.* ‘bench’, ‘table’ and ‘leaves’. This issue makes it difficult to perform image-text matching.

To address this issue, in this work we aim to introduce a classification component to learn more robust latent embeddings. Our motivation is that object labels can typically provide more consistent and less biased information than sentences. As can be seen in Figure 7.1, object labels contain the most important concepts in the image, such as ‘Person’ and ‘Bicycle’ which are commonly mentioned in all of the five sentences. On the other hand, some visual concepts, which are subjectively described in some of the sentences (*e.g.* ‘leaves’ and ‘sweater’) will not appear in the ground-truth labels. Hence, using the object labels as additional supervisory signals is beneficial to correct the biased descriptions and improve the matching between images and texts. Motivated by the mutual complements between matching and classification, we raise the research question **RQ 6: How can we design a unified network for joint multi-modal matching and classification?**

To tackle the question, we propose a unified **Network for joint Multi-modal Matching and Classification** (MMC-Net in Figure 7.2). First, the matching component transforms the input visual and textual features, respectively, via a couple of fully-connected layers and a fusion module. The matching loss is imposed on the outputs of the two fusion modules to maximize their correlation. Then, the classification

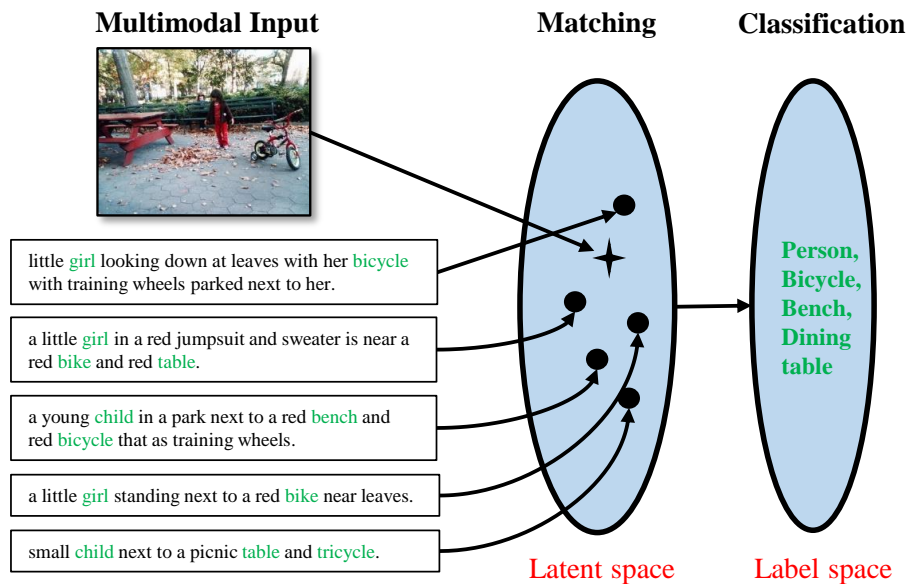


Figure 7.1: Example of joint multi-modal matching and classification. Given one image and its descriptive sentences, they are first co-embedded into a latent space for matching (in red and blue). Then, the visual and textual embedding features are integrated to be a multi-modal representation for classification. In the input sentences, the words related to the ground-truth object labels are in green.

component is built upon the visual and textual embedding features. A compact bi-linear pooling module is used to generate a multi-modal representation vector, based on which the classification loss is computed to predict object labels. In this way, the proposed MMC-Net can jointly learn the latent embeddings and the multi-modal representation in a unified model. On the one hand, the classification component is beneficial to alleviate the biased input, so that the model can learn better robust latent embeddings. On the other hand, the matching component is able to bridge the modality gap between vision and language, and therefore combining visual and textual embedding features can produce a discriminative multi-modal representation for classification.

The contributions of this work are as follows:

- We propose a novel deep multi-modal network (*i.e.* MMC-Net), where the matching and classification components can be seamlessly integrated and help promote each other jointly. MMC-Net is a general architecture that is potentially applicable to diverse multi-modal tasks related to matching and classification.
- We present a multi-stage training algorithm by incorporating the matching and classification loss. It can make the matching and classification components more compatible in a unified model.
- Results on four well-known multi-modal benchmarks demonstrate that MMC-Net outperforms the baseline models that are built for either matching or

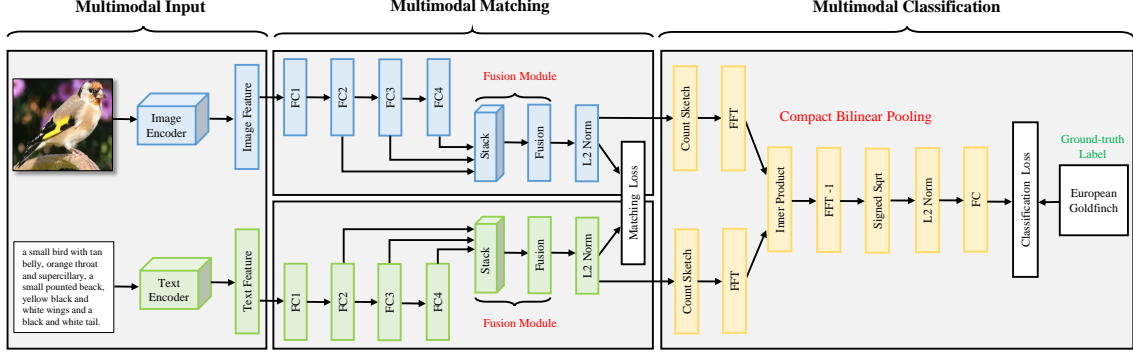


Figure 7.2: The overview architecture of our proposed MMC-Net for joint multi-modal matching and classification. It comprises three key components. (1) The multi-modal input aims to capture visual and textual representations from off-the-shelf encoders (*e.g.* CNN and word2vec). (2) In the matching component, four fully-connected layers in both of the image and text branches are developed to learn the latent embeddings. (3) Based on the visual and textual embedding features, the classification component utilizes a compact bilinear pooling module which can generate a high-order multi-modal representation to perform the prediction. The entire network can be trained with a matching loss and a classification loss.

classification (*i.e.* MM-Net and MC-Net). In addition, our approach achieves competitive performance compared to current state-of-the-art approaches.

The rest of this paper is organized as follows. Section 7.2 introduces the proposed MMC-Net model, and Section 7.3 details its training and inference procedures. Comprehensive experiments in Section 7.4 are used to evaluate the approach. Finally, Section 7.5 concludes the paper and discusses the future work.

7.2 Joint Matching and Classification Network

Overall architecture. Figure 7.2 illustrates the overview architecture of MMC-Net, which mainly consists of three components: multi-modal input, multi-modal matching and multi-modal classification. Given an image and its corresponding text, MMC-Net first utilizes off-the-shelf feature encoders to extract the visual and textual features, respectively. Next, in the multi-modal component, two groups of four fully-connected layers are used in both image and text branches to learn a latent space, where its objective is to minimize the matching loss between the related images and texts. Moreover, the multi-modal classification component is built upon the visual and textual embedding features. We employ a compact bilinear pooling module to generate a high-order and efficient multi-modal representation. The classification loss is computed with respect to the pre-defined ground-truth labels. Next, we will detail each of the three components.

7.2.1 Multi-modal input

In a data collection with N matching image-text pairs, (x_i, y_i) represent the encoded visual and textual features, $i = 1, \dots, N$. Taking these features as input instead of the raw data enables to train the entire network effectively. Also, any common feature encoders are potentially applicable for this network.

Image encoder: we use the powerful CNN model, ResNet-152 [10], which is pre-trained on ImageNet [5]. First, the CNN model is recast to its fully convolutional network (FCN) counterpart, to extract richer region representations. Then we set the smaller side of the image to 512 and isotropically resize the other side. The last max-pooling layer in ResNet-152 is averaged to generate a 2048-dimensional feature vector. Compared with the widely-used VGG feature [7] (*i.e.* 4096-dim), ResNet-152 can provide more discriminative visual representation, while decreasing the feature dimensions (2048 v.s. 4096). The extracted image feature is then fed into the image branch of the matching component.

Text encoder. we employ the simple yet efficient word2vec [188] to represent sentence-level texts. It provides a 300-dimensional feature vector, which is often called Mean vector. Notably, more informative text encoders can be developed based on word2vec, for example the Hybrid Gaussian-Laplacian mixture model (HGLMM) [51] that computes a 18000-dimensional feature vector with 30 centers (*i.e.* $300 \times 30 \times 2$). However, we still use the standard Mean vector due to its high efficiency and low dimensionality. Nevertheless, we clarify that any common text encoders can be potentially adopted to the MMC-Net model.

7.2.2 Multi-modal matching

The multi-modal matching component contains three aspects: latent embedding, fusion module and matching loss.

Latent embedding

As shown in Figure 7.2, the matching component develops two branches of four fully-connected layers to simultaneously project visual and textual features into a discriminative latent space. Note that the parameters of the two branches (drawn in blue and green) are unshared due to the modality specialization. The channels from FC1 to FC4 are set to $\{2048, 512, 512, 512\}$ in both of the two branches. First, the input visual and textual features are normalized with the batch normalization (BN) [135]. Then FC1 is regularized by a dropout layer with 0.5 probability, and instead other fully-connected layers are regularized with the BN layer. ReLU is used after the fully-connected layers.

Fusion module

Exploiting multi-layer features has been well-studied in many deep neural networks [18, 26, 31, 107], as it allows to take advantage of different levels of hidden representations in the networks. Driven by this, we introduce a fusion module to generate a multi-layer embedding feature. Since the FC2, FC3 and FC4 layers have the same number of channels, it is feasible to stack their feature vectors together. Then we employ a convolutional operation to learn adaptive weights while fusing the three layers.

We denote the stack layer in the two branches as $S(x_i)$ and $S(y_i)$, respectively. The stack layer, a 512×3 matrix, is convolved by the convolutional filter, which has a size of $1 \times 1 \times 3$. Note that, the three weights are shared over the spatial dimensions of the stack layer. We can compute the fused visual feature $f(x_i)$ and textual feature $g(y_i)$ by

$$f(x_i) = W_I^{fuse} \odot S(x_i) + b_I^{fuse}, \quad (7.1)$$

$$g(y_i) = W_T^{fuse} \odot S(y_i) + b_T^{fuse}, \quad (7.2)$$

where W_I^{fuse} and W_T^{fuse} are the fusion weights to be learned (*i.e.* 3 elements) b_I^{fuse} and b_T^{fuse} are the bias vectors (*i.e.* 512 elements). The operator \odot represents the convolutional operation.

Although the common element-wise operators such as sum-pooling and inner product are simple to compute, they do not adapt the importance of different layers. Another fusion approach is concatenating the three 512-Dim vectors into one 3×512 -Dim vector. However, the concatenation output will increase the feature dimensionality and make it more expensive to compute the matching loss. To summarize, the convolutional fusion module can provide marked performance improvements, while it has a minimal increase to the total parameters used in the network.

Matching loss

As a common practice, the matching distance between $f(x_i)$ and $g(y_i)$ is computed with the cosine distance [52, 53, 76]

$$d(f(x_i), g(y_i)) = 1 - \frac{f(x_i) \cdot g(y_i)}{\|f(x_i)\| \cdot \|g(y_i)\|}. \quad (7.3)$$

Smaller distances indicate more similar image-text pairs. Both $f(x_i)$ and $g(y_i)$ are L2-normalized before computing their cosine distance. To preserve the similarity constraints in the latent space, we define the matching loss based on an efficient bi-directional rank loss function, similar to [53, 181, 214]. The loss function needs to handle the two triplets, $(x_i, y_i, y_{i,k}^-)$ and $(y_i, x_i, x_{i,k}^-)$, where $x_{i,k}^- \in X_i^-$ and $y_{i,k}^- \in Y_i^-$

are the negative images and texts, $k = 1, \dots, K$. To exploit more representative non-matching pairs, we pick the top K most dissimilar candidates in each mini-batch. Intuitively, this loss function is designed to decrease the distances of matching pairs (e.g. x_i and y_i) and increase the distances of non-matching pairs (e.g. x_i and $y_{i,k}^-$, y_i and $x_{i,k}^-$). Formally, the matching loss based on the fused features is:

$$\begin{aligned} \mathcal{L}_{mat}^{fuse} = \sum_{i=1}^N \sum_{k=1}^K \max & \left[0, d(f(x_i), g(y_i)) - d(f(x_i), g(y_{i,k}^-)) + m \right] \\ & + \alpha \max \left[0, d(f(x_i), g(y_i)) - d(f(x_{i,k}^-), g(y_i)) + m \right], \end{aligned} \quad (7.4)$$

where m is a margin parameter, and α is used to balance the importance of the two triplets. Minimizing this loss cost will lead to a desirable latent space, where the matching distance $d(f(x_i), g(y_i))$ should be smaller than any of the non-matching ones $d(f(x_i), g(y_{i,k}^-))$ and $d(f(x_{i,k}^-), g(y_i))$, $\forall x_{i,k}^- \in X_i^-$, $\forall y_{i,k}^- \in Y_i^-$.

In Figure 7.3, we make use of the t-SNE algorithm [207] to visualize our embedding features (i.e. $f(x_i)$ and $g(y_i)$). We use the 1,000 images and 5,000 texts from the MSCOCO test set. It can be seen that in the distribution map an image feature (in red) is properly surrounded by several related text features (in green), as each image is annotated by five ground-truth matching texts in the dataset. Therefore, this visualization shows that our embedding model can align the images and texts due to learning their semantic correlation. In addition, some images and texts corresponding to the points are shown in the windows. We can see that the embeddings can cluster related images and texts together.

7.2.3 Multi-modal classification

The classification component aims to incorporate the visual and textual embedding features and then generates a multi-modal representation for predicting object labels. In the following, we detail the classification component including a bilinear pooling module and classification loss.

Bilinear pooling

We take advantage of a bilinear pooling module to incorporate visual and textual embedding features learned in the matching component. The bilinear pooling [215] aims to model the pair-wise multiplicative intersection between all elements of two vectors. It can generate more expressive features than other basic operators such as element-wise sum or product. The standard bilinear pooling is formulated with

$$\mathcal{B}(x_i, y_i) = f(x_i)^T g(y_i), \quad (7.5)$$

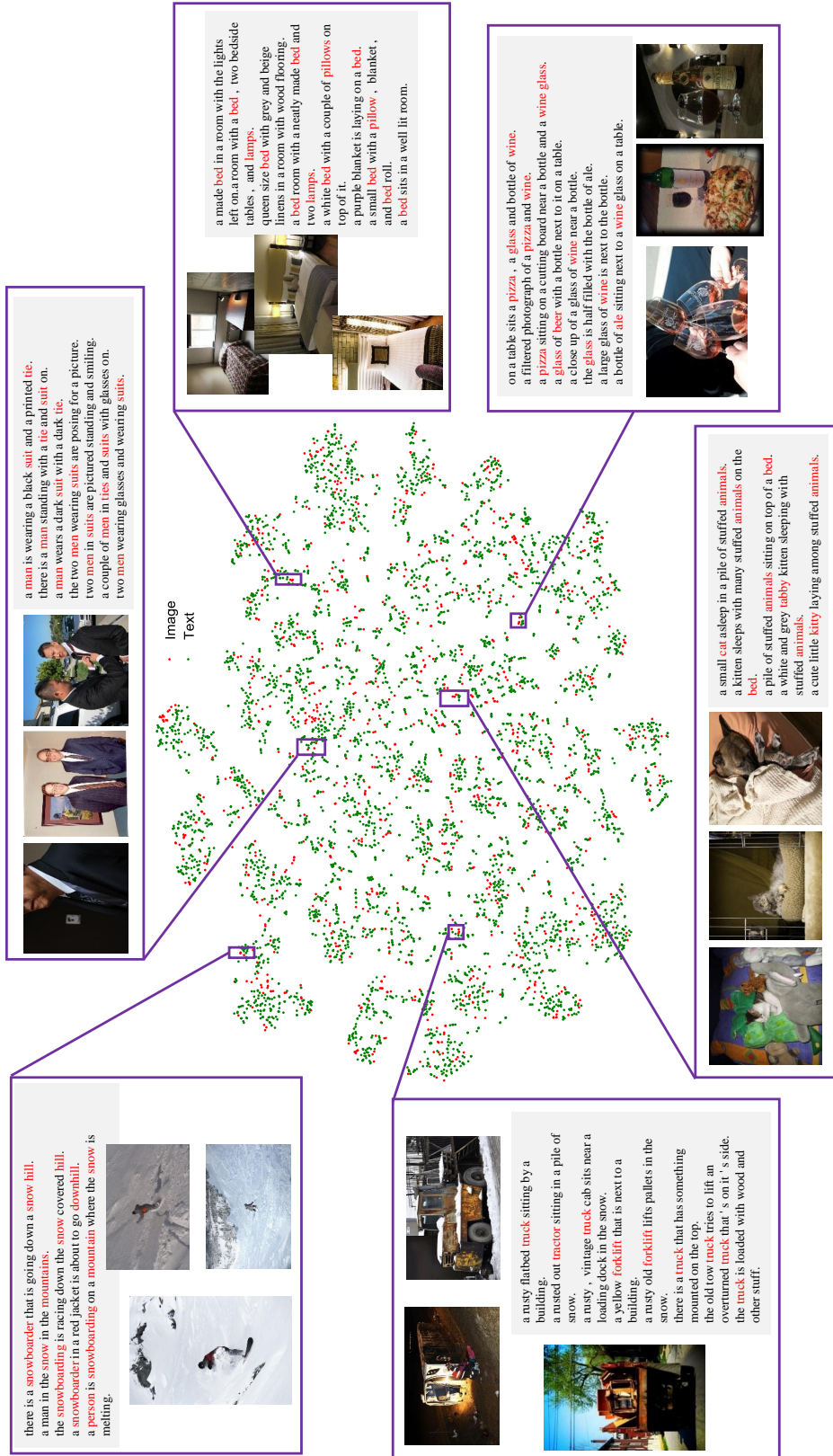


Figure 7.3: Visualization of the visual and textual embedding features learned in the matching component. Each image (in red) is related to several corresponding texts (in green). We present some images and texts corresponding to the points in the distribution map. Some semantic words are highlighted in red.

Algorithm 2: CBP with latent embedding features

- 1: **Input:** $f(x_i) \in \mathbb{R}^M$, $g(y_i) \in \mathbb{R}^M$
 - 2: **Output:** $\mathcal{B}(x_i, y_i) \in \mathbb{R}^D$
 - 3: **Initialize hash functions:** h_1, s_1, h_2, s_2
 - For** $j \leftarrow 1 \cdots M$
 - sample $h_1[j], h_2[j]$ from $\{1, \dots, D\}$
 - sample $s_1[j], s_2[j]$ from $\{-1, 1\}$
 - End for**
 - 4: **Compute count sketches:**
 - $\hat{f}(x_i) = [0, \dots, 0]$, $\hat{g}(y_i) = [0, \dots, 0]$
 - For** $j \leftarrow 1 \cdots D$
 - $\hat{f}(x_i)[h_1[j]] = \hat{f}(x_i)[h_1[j]] + s_1[j] \cdot f(x_i)[j]$
 - $\hat{g}(y_i)[h_2[j]] = \hat{g}(y_i)[h_2[j]] + s_2[j] \cdot g(y_i)[j]$
 - End for**
 - 5: **Convolution of Count Sketches:**
 - $\mathcal{B}(x_i, y_i) = \text{FFT}^{-1}(\text{FFT}(\hat{f}(x_i)) \circ \text{FFT}(\hat{g}(y_i)))$,
 - where the \circ denotes element-wise multiplication.
-

Since $f(x_i)$ and $g(y_i)$ are $1 \times M$ vectors (*i.e.* $M = 512$), $\mathcal{B}(x_i, y_i)$ becomes an $M \times M$ matrix that is then reshaped to be a $1 \times M^2$ vector. Due to the high dimensionality of the bilinear vector (*i.e.* M^2), we instead use the compact bilinear pooling (CBP) variant [216], which can decrease the dimensionality to D (where $D \ll M^2$) while retaining the strong discrimination. In contrast to [216, 217] in which they simply perform the CBP module with the input visual or textual features, we build the CBP module based on the latent embeddings to generate a multi-modal feature vector (Figure 7.2).

The computational procedure of the CBP module is detailed in Algorithm 2. At first, we initialize several hashing functions from the pre-defined sets. Then, it computes the count sketches [218] to maintain linear projections of a vector with several random vectors. Finally, we make use of the Fast Fourier Transformation (FFT) to compute the convolution of the count sketches, and produce a bilinear vector $\mathcal{B}(x_i, y_i)$ by an inverse FFT. The count sketches have the properties:

$$E[\langle \hat{f}(x_i), \hat{g}(y_i) \rangle] = \langle f(x_i), g(y_i) \rangle, \quad (7.6)$$

$$\text{Var}[\langle \hat{f}(x_i), \hat{g}(y_i) \rangle] \leq \frac{1}{D}(\langle f(x_i), g(y_i) \rangle^2 + \|f(x_i)\|^2 + \|g(y_i)\|^2). \quad (7.7)$$

Next, the bilinear vector $\mathcal{B}(x_i, y_i)$ is processed by a signed square-root layer and an L2 normalization layer. Then, we employ a fully-connected layer to estimate the prediction. Assume that there are C object labels pre-defined in the dataset, the

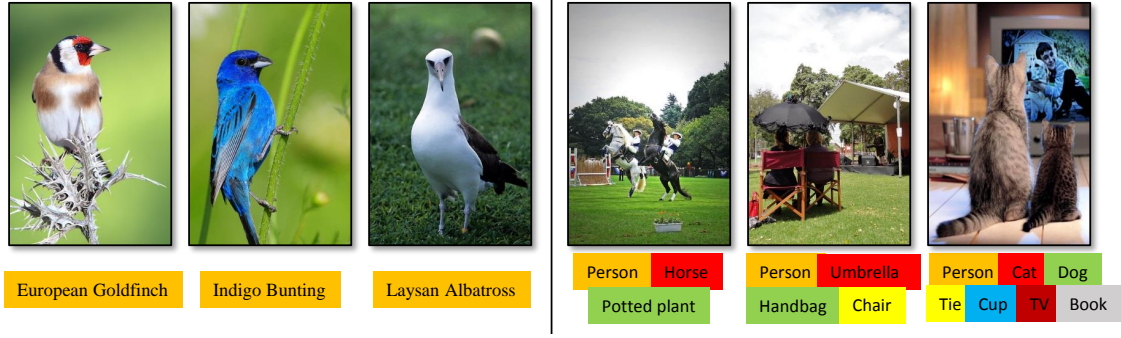


Figure 7.4: Left: Examples of single-label images from CUB-Bird [139]. Right: Examples of multi-label images from MSCOCO [117]. The ground-truth labels are shown under the images.

j -th class probability is predicted with

$$a_{i,j} = \sum_{k=1}^D W_{j,k} \mathcal{B}(x_i, y_i)_k, j = 1, \dots, C. \quad (7.8)$$

where $W_{j,k}$ is the parameter matrix with the size of $D \times C$. For simplicity, we do not show the signed square-root and the L2 normalization in this formulation.

Classification loss

The objective of the classification component is to minimize the loss cost of the prediction with respect to the given ground-truth labels. Figure 7.4 shows some images that are annotated by single label or multiple labels. We need to utilize different loss functions for single-label and multi-label classification, respectively.

1) *Single-label classification.* For example, the fine-grained classification in the left of Figure 7.4, each image is labelled with a fine bird category. To train the classification component, we use the softmax loss function

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C \delta(g_i = j) \log p_{i,j}, \quad (7.9)$$

$$p_{i,j} = \frac{\exp(a_{i,j})}{\sum_{k=1}^C \exp(a_{i,k})}, \quad (7.10)$$

where g_i is the ground-truth label corresponding to x_i . $\delta(g_i = j)$ is 1 when $g_i = j$, otherwise is 0.

2) *Multi-label classification.* As shown in the right of Figure 7.4, images annotated with multiple labels can provide richer information about the visual content. Although many of these labels may appear in the input text, they can still offer

complementary labels which are ignored in the text due to less visual attention. We employ the sigmoid cross-entropy loss function to supervise the multi-label classification. The total cost sums up K of element-wise loss terms

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C g'_{i,j} \log p'_{i,j} + (1 - g'_{i,j}) \log(1 - p'_{i,j}), \quad (7.11)$$

$$p'_{i,j} = \frac{1}{1 + \exp(-a_{i,j})}, \quad (7.12)$$

where $g'_{i,j} \in \{0, 1\}$ is the ground-truth label indicating the absence or presence of the j -th class.

7.3 Training and Inference

This section describes the training procedure of the MMC-Net model. Also, we present the inference manner for multi-modal matching and classification.

Multi-stage training procedure

The optimization objective in the model is to minimize the total training loss which merges the matching and classification loss together

$$\min_W \mathcal{L}_{total} = \mathcal{L}_{mat} + \beta \mathcal{L}_{cls}, \quad (7.13)$$

where the parameter β is used to regulate the two loss terms. The parameters W in the network mainly contains W_I and W_T in the image and text branches, and W_{CBP} in the compact bilinear pooling module.

We propose a multi-stage training algorithm to better model the matching and classification components. As summarized in Algorithm 3, the training procedure consists of three stages. During the first stage, we train the matching component with the loss \mathcal{L}_{mat} . For the second stage, we use the loss \mathcal{L}_{cls} to train the parameters in the classification component. In this stage, only the parameters in the classification component are updated while the parameters in the matching component are all frozen. In the third stage, the model is initialized by the parameters learned in the first and second stages. It aims to jointly fine-tune the whole network based on the total loss \mathcal{L}_{total} . Due to using this multi-stage fashion, it is feasible to promote the training of the entire network and maintain the high performance.

Inference procedure

We present the inference procedure for multi-modal matching and classification.

Algorithm 3: Multi-stage Training Algorithm for MMC-Net.

- 1: **The first stage:** train the matching component.
 initialize: learning rate λ_1 , training iterations T_1 , $t = 0$.
while $t < T_1$ **do**
 $t \leftarrow t + 1$
 compute the matching loss \mathcal{L}_{mat} in Eq.(7.4);
 update the parameters in the image and text branches:
 $W_I^{(t)} = W_I^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{mat}}{\partial W_I^{(t-1)}}$;
 $W_T^{(t)} = W_T^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{mat}}{\partial W_T^{(t-1)}}$;
end while
 - 2: **The second stage:** train the classification component.
 initialize: learning rate λ_2 ($< \lambda_1$), training iterations T_2 , $t = 0$.
while $t < T_2$ **do**
 $t \leftarrow t + 1$
 compute the classification loss \mathcal{L}_{cls} in Eq.(7.9) or Eq.(7.11);
 update the parameters in the compact bilinear pooling module:
 $W_{CBP}^{(t)} = W_{CBP}^{(t-1)} - \lambda_2^{(t)} \frac{\partial \mathcal{L}_{cls}}{\partial W_{CBP}^{(t-1)}}$;
end while
 - 3: **The third stage:** jointly fine-tune the whole network.
 initialize: learning rate λ_3 ($< \lambda_2$), training iterations T_3 , $t = 0$.
while $t < T_3$ **do**
 $t \leftarrow t + 1$
 compute the total loss in Eq.(7.13);
 update all the parameters in the network:
 $W_I^{(t)} = W_I^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_I^{(t-1)}}$;
 $W_T^{(t)} = W_T^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_T^{(t-1)}}$;
 $W_{CBP}^{(t)} = W_{CBP}^{(t-1)} - \lambda_2^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_{CBP}^{(t-1)}}$;
end while
-

(1) Multi-modal matching: For the image-to-text matching, given a query image x_q , its purpose is to search for relevant texts *w.r.t.* x_q from a text database Y . Likewise, the text-to-image matching aims to retrieve related images from an image database X , given a query text y_q . In the MMC-Net model, the fused visual and textual features learned in the fusion module are used to compare the matching distance, denoted as $d(f(x_q), g(y_i))$ or $d(f(x_i), g(y_q))$, where $y_i \in Y, x_i \in X$. The k -nearest neighbor (k -NN) search is used to find the top- k most similar candidates.

(2) Multi-modal classification: Its inference is based on the probabilities predicted by the last fully-connected layer in the classification component. For the single-label case, the element that has the maximum probability corresponds to the predicted class. As for the multi-label case, the items whose probabilities in the prediction are more than 0.5 are estimated to contain the corresponding object classes.

Table 7.1: Summary of four multi-modal datasets used in the experiments. TPI indicates the number of matching Texts Per Image.

Dataset	#Total	#Category	#Training	#Test	#TPI
Pascal Sentence	1,000	20	800	100	5
MSCOCO	~120K	80	82,783	1,000	5
Flowers	8,189	102	2,040	6,149	10
CUB-Bird	11,788	200	5,994	5,794	10

7.4 Experiments

In this section, we evaluate the performance of the proposed MMC-Net on four well-known multi-modal benchmarks. We first introduce the configuration in the experiments, including the datasets, evaluation metrics, parameter settings and baseline models. Then we assess the performance of MMC-Net for tasks of multi-modal matching and classification and compare its results with those of the baseline models. Furthermore, we conduct the ablation study to fully analyze MMC-Net. Lastly, we compare our results with the state-of-the-art approaches.

7.4.1 Experimental setup

Dataset protocols

We perform the experiments on four well-known multi-modal datasets. Some image and text examples are shown in Figure 7.5.

Pascal Sentence [219]. It contains 1,000 images from 20 categories (50 images per category), and one image is described by five different sentences. We pick 800 images for training (40 images per category), 100 images for validation (5 images per category), and 100 images for test (5 images per category). In total, there are $40 * 20 * 5 = 4,000$ image-text training pairs, $5 * 20 * 5 = 500$ validation pairs, and $5 * 20 * 5 = 500$ test pairs.

MSCOCO [117]. It includes 82,783 training images and 40,504 validation images in total. We pick five descriptive sentences for one image and generate $82,783 * 5 = 413,915$ training pairs. For a fair comparison, we use the same 1,000 test images used in recent works [52, 53, 76].

Flowers [138]. This dataset [138] contains 102 classes with a total of 8,189 images. 2,040 images (train+val) are used in the training stage and the rest 6,149 images are for testing. Reed *et al.* [195] collected fine-grained visual descriptions for these images by using the Amazon Mechanical Turk (AMT) platform. One image is described by ten sentence-level descriptions. Therefore, we can obtain $2040 * 10 = 20,400$ training pairs and $6149 * 10 = 61,490$ testing pairs.

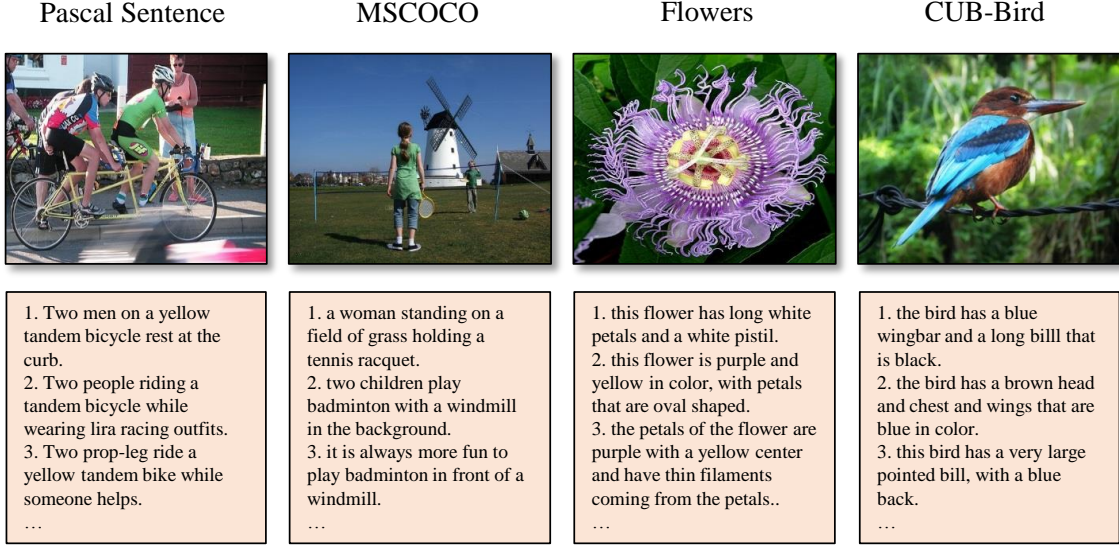


Figure 7.5: Example of four multi-modal datasets. Several textual descriptions are listed for each image.

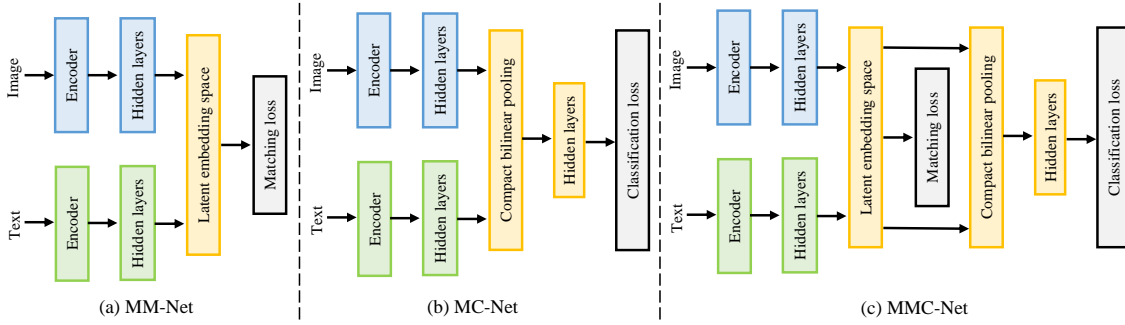


Figure 7.6: Conceptual illustration of three multi-modal networks. (a) Multi-modal Matching Network. (b) Multi-modal Classification Network. (c) Multi-modal Matching and Classification Network. Note that, the parameters in the image and text branches are unshared, as drawn in blue and green.

CUB-Bird [139]. It contains 11,788 bird images from 200 categories. 5,994 images are for training, and 5,794 images are for testing. Similarly, ten sentences are provided to describe one image [195]. As a result, it has $5994 * 10 = 59,940$ pairs for training, and $5794 * 10 = 57,940$ pairs for testing.

Evaluation Metrics

We evaluate the performance of multi-modal matching and multi-modal classification, separately. (1) For multi-modal matching, We employ the widely-used retrieval metric $R@K$, which is the recall rate of a correctly retrieved ground-truth at top K candidates (e.g. $K = 1, 5, 10$) [55, 190]. It includes results of both image-to-text ($I \rightarrow T$) and text-to-image retrieval ($T \rightarrow I$). (2) Considering multi-modal classification, We compute the Top-1 classification accuracy for Pascal Sentence, Flowers and

CUB-Bird. Since MSCOCO is a multi-label classification dataset, we evaluate the performance on it using the average precision with the average precision (AP) across multiple classes.

Implementation details

We implemented the proposed approach based on the publicly available Caffe library [130]. It is important to shuffle the training samples randomly during the data preparation stage. The hyper-parameters were evaluated on the validation set of each dataset. For instance, we set $\alpha = 2$ and $m = 0.1$ while computing the matching loss function on all the datasets. The number of non-matching pairs in the negative sets was $K = 20$ for Pascal Sentence, Flowers and CUB-Bird, and $K = 50$ for MSCOCO. We used a mini-batch size of 128 for Pascal Sentence, Flowers and CUB-Bird, and 1500 for MSCOCO. Note that, we use a larger K and mini-batch size for MSCOCO, because it has enormously more training samples, compared to the other three datasets. We trained the model using SGD with a weight decay of 0.0005, a momentum of 0.9. The learning rate was initialized with 0.1 and was divided by 10 when the loss stopped decreasing.

Baseline Models.

To verify the effectiveness of the proposed MMC-Net, we implemented two baseline models. (1) **MM-Net**: a baseline model for multi-modal matching as illustrated in Figure 7.6(a). It only contains the matching component of the MMC-Net (Figure 7.2), which is trained with the matching loss. (2) **MC-Net**: a baseline model for multi-modal classification as illustrated in Figure 7.6(b). It has the similar architecture as the MMC-Net, however, it does not compute the matching loss between visual and textual features. MC-Net is only trained with the classification loss.

7.4.2 Results on multi-modal retrieval

We conduct the cross-modal retrieval experiments on the four datasets. To verify the effectiveness of adding a classification component in MMC-Net, we use the baseline MM-Net for comparison. Table 7.2 and Table 7.3 report the results of image-to-text and text-to-image retrieval, respectively. Overall, MMC-Net can achieve considerable improvements over MM-Net for both I→T and T→I retrieval. These results reveal that the classification component in MMC-Net can help in improving the learning of embedding features in the matching component. Moreover, we can observe more insights from these results as follows:

7. JOINT MATCHING AND CLASSIFICATION

Table 7.2: Image-to-text retrieval results compared between MMC-Net and MM-Net. The proposed MMC-Net can outperform the baseline MM-Net with considerable gains across all the four datasets.

Method	Pascal Sentence			MSCOCO			Flowers			CUB-Bird		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	47.0	85.0	92.0	55.5	84.2	91.4	58.1	82.5	88.5	32.5	61.4	72.5
MMC-Net	52.0	87.0	93.0	57.0	85.8	92.7	78.7	93.9	96.0	39.2	66.9	76.4

Table 7.3: Text-to-image retrieval results compared between MMC-Net and MM-Net. Compared to MM-Net, MMC-Net can achieve better retrieval results.





Method	Pascal Sentence			MSCOCO			Flowers			CUB-Bird		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	38.4	80.6	88.6	44.7	79.5	89.5	32.7	46.4	52.9	18.3	25.6	28.8
MMC-Net	41.0	81.2	92.5	46.2	80.8	90.5	43.6	54.8	58.6	25.8	31.4	34.5

- By comparison with MM-Net, MMC-Net yields more performance gains on Flowers and CUB-Bird than Pascal Sentence and MSCOCO. For example, the performance gap between MMC-Net and MM-Net is below 5% on Pascal Sentence and MSCOCO, but above 5% on Flowers and CUB-Bird across all the measurements. One reason is that both Flowers and CUB-Bird are fine-grained datasets, and the textual descriptions cannot fully represent the discrimination among different samples. Hence, the results of MM-Net are limited on these two datasets. Instead, MMC-Net can make use of fine-grained class labels to enhance the discriminative abilities when matching images and texts.
- The results of $T \rightarrow I$ retrieval are lower than those of the $I \rightarrow T$ retrieval on the four datasets. This is because each image can retrieve several related textual descriptions, but one text is corresponded to only one matched image. We believe that refining the datasets is a favorable solution to narrow the performance gap between the $I \rightarrow T$ and $T \rightarrow I$ retrieval.
- For Flowers and CUB-Bird, their results are still not satisfactory, especially for the $T \rightarrow I$ retrieval. Currently, the fine-grained multi-modal matching still remains challenging, but it is a promising research direction in the field.

































In addition, we present the qualitative retrieval results as shown in Figure 7.7. We can observe that MMC-Net obtains better retrieved candidates than MM-Net, for both $I \rightarrow T$ and $T \rightarrow I$ retrieval. Furthermore, we visualize the visual and textual embedding features learned in the matching component of MMC-Net. As mentioned earlier in 7.3, it has shown the embedding map with the MSCOCO test set.

7.4.3 Results on multi-modal classification

Next, we conduct the multi-modal classification experiments on the datasets. To demonstrate the benefit of using a matching component for classification, we compare the MMC-Net model with the baseline MC-Net model. Table 7.4 reports the

Query Image	MM-Net: Retrieved texts	MMC-Net: Retrieved texts
Pascal Sentence 	1. People riding tandem bicycle. 2. Two prop-leg ride a yellow tandem bike while someone helps. 3. Young man wearing jeans and helmet rides his motorcycle in front of a small crowd. 4. A man wearing a helmet does a wheelie on a motorcycle as a crowd watches.	1. Two prop-leg ride a yellow tandem bike while someone helps. 2. People riding tandem bicycle. 3. Two people riding a tandem bicycle while wearing lira racing outfits. 4. Young man wearing jeans and helmet rides his motorcycle in front of a small crowd.
MSCOCO 	1. a man putting together a kite on the floor of a room. 2. man folding banner while holding stick in unfinished carpet. 3. a man folding a giant paper airplane on the floor. 4. a tiny toddler carries a giant bookbag and bag.	1. a man putting together a kite on the floor of a room. 2. man folding banner while holding stick in unfinished carpet. 3. a man folding a giant paper airplane on the floor. 4. a man inside a room putting together a white kite.
Flowers 	1. this flower is pink and white in color, with petals that have pink veins. 2. this pink flower has several filaments sticking out of the receptacle. 3. this flower has pale pink petals with veins and a white center. 4. this flower has petals that are pink with long stamen.	1. this flower is pink and white in color, with petals that have pink veins. 2. this flower has pale pink petals with veins and a white center. 3. this flower has very light pink petals that have darker pink veins, a yellow ovary, and white stamen. 4. this pink flower has several filaments sticking out of the receptacle.
CUB-Bird 	1. a dark brown beak with a long beak and large wingspan. 2. this bird has a dark grey color, with a large bill and long wingspan. 3. this dull colored bird is brown all over, has large wings and a long large bill. 4. a bird with a large, hooked bill, white superciliary and cheek patch, brown crown, and brown body.	1. a dark brown beak with a long beak and large wingspan. 2. large bird that is complete brown, with white stripes littering it's wings and a long blunted bill. 3. a bird with a large, hooked bill, white superciliary and cheek patch, brown crown, and brown body. 4. this dull colored bird is brown all over, has large wings and a long large bill.

(a) Image-to-text retrieval

Query Text	MM-Net: Retrieved images	MMC-Net: Retrieved images
Pascal Sentence An Swiss-Air flight has just taken off from a runway.	   	   
MSCOCO a woman in white shirt holding bananas next to door.	   	   
Flowers the bright orange petals are highlighted by brown spots and the prominent stamen are topped with dark brown anthers.	   	   
CUB-Bird this bird is light brown, has a long hooked bill, and looks dumb.	   	   

(b) Text-to-image retrieval

Figure 7.7: Image-text retrieval examples on the datasets. For (a) image-to-text retrieval, the ground-truth matching texts are in green. For (b) text-to-image retrieval, the red number in the upper left corner of one image is the ranking order, and the green frame corresponds to the ground-truth matching image. For the I→T and T→I retrieval, MMC-Net can retrieve more accurate candidates than MM-Net.

classification results, where MMC-Net achieves consistent improvements over MC-Net across all the four datasets. It shows that the matching component is able to promote the classification component due to combining the embedding features to generate more discriminative multi-modal representations. Also, MMC-Net has a generalization ability for different types of classification datasets, including either natural images or fine-grained images.

7. JOINT MATCHING AND CLASSIFICATION





	Pascal Sentence	MSCOCO	Flowers	CUB-Bird
	 <div data-bbox="446 257 550 421">A striped sofa and office chairs are near a ping pong table.</div>	 <div data-bbox="692 257 796 421">a tennis player wiping his face off with a towel.</div>	 <div data-bbox="922 257 1051 421">the petals of the flower are purple in color and have green stems with green sepals.</div>	 <div data-bbox="1190 257 1319 421">a bird with a medium yellow bill, white body webbed feet and gray wings.</div>
MC-Net	1. chair 2. tv/monitor 3. sofa 4. diningtable 5. bottle	1. person 2. chair 3. sports ball 4. tennis racket 5. dining table	1. bolero deep blue 2. garden phlox 3. canterbury bells 4. bougainvillea 5. snapdragon	1. Glaucous winged Gull 2. Ring billed Gull 3. California Gull 4. Herring Gull 5. Heermann Gull
MMC-Net	1. sofa 2. chair 3. Diningtable 4. tv/monitor 5. potted plant	1. person 2. tennis racket 3. chair 4. bench 5. sports ball	1. canterbury bells 2. bolero deep blue 3. foxglove 4. stemless gentian 5. garden phlox	1. Herring_Gull 2. California_Gull 3. Western_Gull 4. Ring_billed_Gull 5. Slaty_backed_Gull

Figure 7.8: Multi-modal classification examples on the datasets. Given an input image-text pair, the Top-5 predictions are estimated based on MC-Net and MMC-Net. The ground-truth classes are in green. By comparison, MMC-Net obtains more accurate predictions than MC-Net.

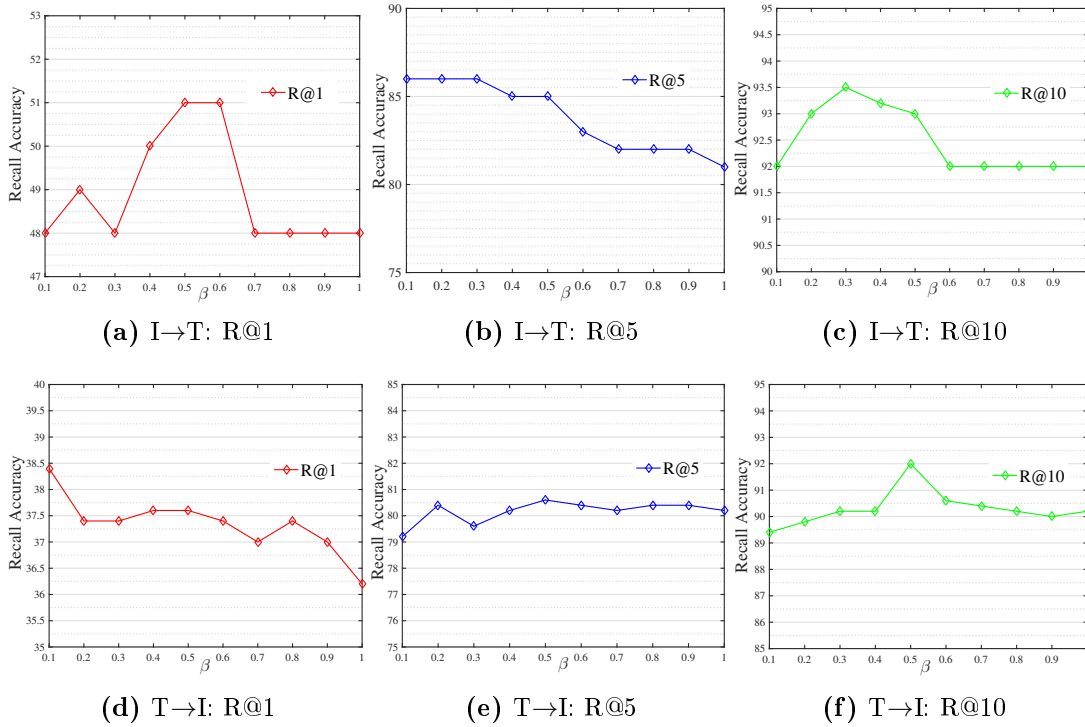


Figure 7.9: Effect of the parameter β on the performance of MMC-Net. The retrieval results on Pascal Sentence are reported. We select $\beta = 0.5$ by comparing these results.

7.4.4 Parameter analysis

Next, we aim to analyze the effects of three key parameters in MMC-Net.

Table 7.4: Comparison of the multi-modal classification accuracy between MMC-Net and MC-Net. For the four datasets, MMC-Net can outperform MC-Net with consistent performance gains.

Method	Pascal Sentence	MSCOCO	Flowers	CUB-Bird
MC-Net	71.0	77.6	94.0	80.7
MMC-Net	74.0	79.3	95.2	82.4

Table 7.5: Effect of the mini-batch size on the performance of MMC-Net. We train the model with different mini-batch sizes and compare their retrieval results on MSCOCO.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
batch size=100	42.5	74.6	87.4	36.6	73.8	86.8
batch size=250	52.6	83.3	91.7	43.0	79.5	89.4
batch size=500	56.6	85.3	92.7	46.0	80.5	90.1
batch size=1000	56.2	85.8	93.0	46.5	80.5	90.1
batch size=1500	57.0	85.8	92.7	46.2	80.8	90.5
batch size=2000	56.7	85.5	92.8	46.7	80.6	90.4

Effect of the mini-batch size.

Since the loss function for multi-modal matching aims to search for hard negative samples, it is essential to define a large mini-batch to increase the search space. For example, we selected a mini-batch size of 1500 for MSCOCO due to its large-scale data. To study the effect of varying different batch sizes, we used different batch sizes to train MMC-Net and tested their performance. Considering the number of negative pairs in each mini-batch is $K = 50$ for MSCOCO, we varied the batch size with 100, 250, 500, 1000, 1500 and 2000. Table 7.5 compares the retrieval results on MSCOCO with different batch sizes. We can observe that the performance is low when the batch size is 100. By increasing the size to 500, it can achieve significant gains across all the measurements. We further raise the size to 2000, however there is no important influence on the results. Finally, we select batch size=1500 due to its slightly superior results.

Effect of the parameter β .

Recall that MMC-Net is trained by integrating the matching and classification loss, we use the parameter β to balance the weights of the two loss functions as defined in Eq. 7.13. This experiment aims to analyze the effect of β on the performance. Figure 7.9 shows the cross-modal retrieval results on Pascal Sentence. The R@1, R@5 and R@10 results are shown separately, when β varies from 0.1 to 1. We pick $\beta = 0.5$ by fully comparing these results.

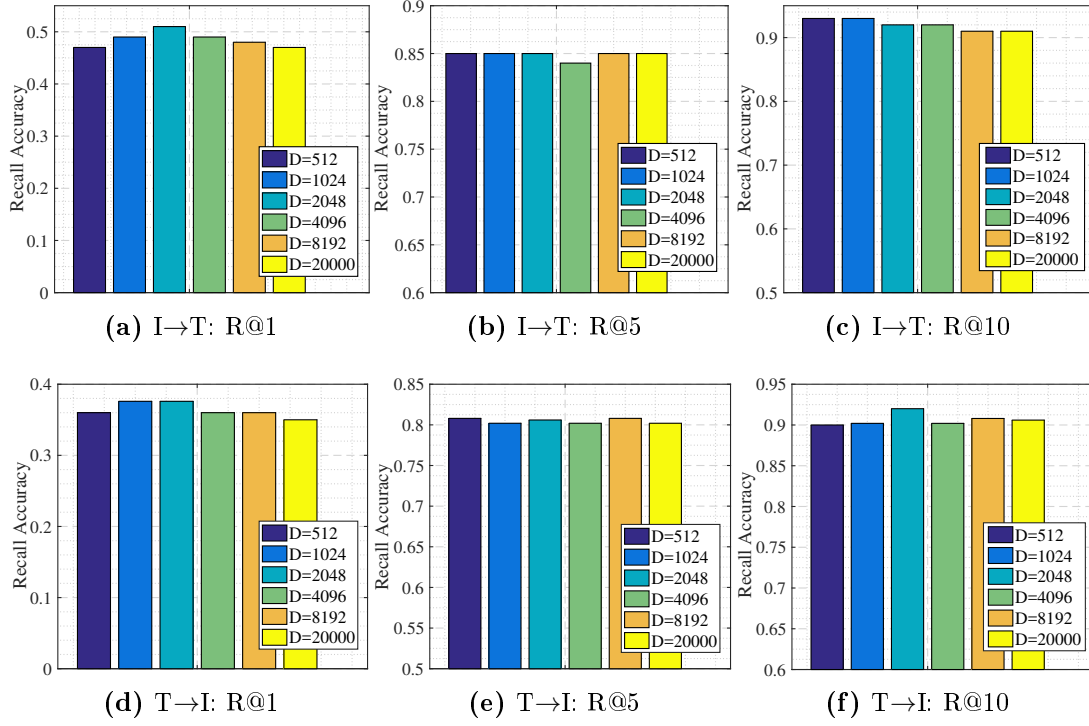


Figure 7.10: Effect of the parameter D on the performance of MMC-Net. We present the retrieval results on Pascal Sentence by using different sizes of D . We select $D = 2048$ that can bring better results.

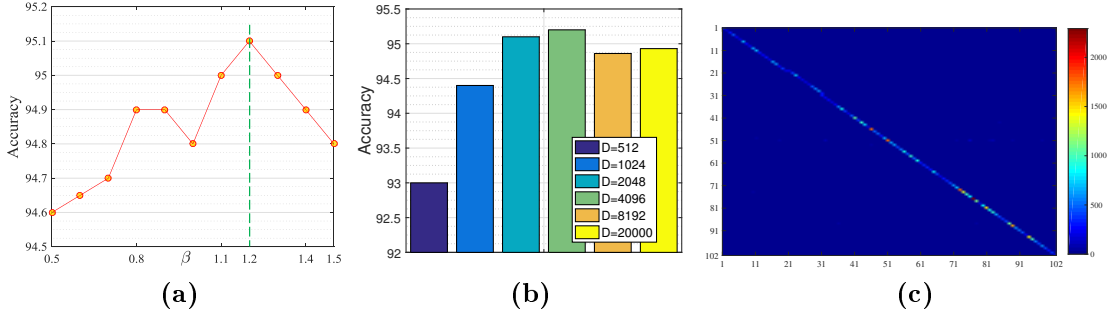


Figure 7.11: Effect of the parameters on the performance of MMC-Net. We report the Top-1 classification results on Flowers. (a) Analysis of the parameter β . (b) Analysis of the parameter D . (c) Confusion matrix of 102 Flowers classes. The diagonal line demonstrates the high accuracy per flower class.

Effect of the parameter D .

In the classification component, a CBP module can integrate visual and textual embedding features into a D -dimension multi-modal vector. In this experiment, we analyze D with $\{512, 1024, 2048, 4096, 8192, 20000\}$, which are all significantly lower than the original bilinear pooling vector (*i.e.* $512 \times 512 = 262,144$). In Figure 7.10, we present the compared results on Pascal Sentence. When $D = 2048$, MMC-Net can achieve better results compared to others.

Since MSCOCO is also composed of scene images like Pascal Sentence, it is straightforward and general to employ the same parameters β and D . In contrast, Flowers and CUB-Bird are commonly used for fine-grained recognition. It is needed to evaluate their parameters separately for Pascal Sentence and MSCOCO. To this end, we estimated the effects of the parameters on the classification accuracy of Flowers, and then applied the same parameters to CUB-Bird for generalization. Figure 7.11 presents the analysis of parameters on Flowers. As for the parameter β shown in Figure 7.11a, the best precision accuracy is achieved with 95.1% for $\beta = 1.2$. As shown in Figure 7.11b, the accuracy is maximized (*i.e.* 95.2%) when $D = 4096$. In the experiments, we set $\beta = 1.2$ and $D = 4096$ for Flowers and CUB-Bird. Additionally, we show the confusion matrix of 102 Flowers categories in Figure 7.11c.

7.4.5 Component analysis

Furthermore, we show ablation study to provide in-depth analysis.

Analysis of the fusion module

This test aims to verify the effectiveness of using the fusion module in the matching component. We build a convolutional fusion module in MMC-Net, which can also be applied on the baseline MM-Net. In Table 7.6, we report the results for both MMC-Net and MM-Net on the Pascal Sentence test set. We can see that using a fusion module can bring considerable performance improvements on all R@K measurements by considerable improvements, compared to the counterparts without using any fusion module. For an additional comparison, we further implement two simple fusion modules: element-wise sum and multiplication. Their results are inferior to those of the convolutional fusion, because they do not consider the weights of different layers. Instead, the convolutional fusion can learn adaptive weights to produce a superior fused feature while spending only three parameters. All the weights can be learned dynamically and adaptively with other network parameters without any manual tuning.

Analysis of the CBP module

We conduct this experiment to test the use of the CBP module in MMC-Net. For comparison, we present two other methods to integrate the visual and textual features. The first method starts by the concatenation of the two features to construct a multi-modal representation and then feed it into a fully-connected (FC) layer to perform the classification. The second one is using the traditional bilinear pooling (BP) to produce a high-order multi-modal representation. Table 7.7 reports the compared results of different classification modules. The model with CBP can

Table 7.6: Analysis of the fusion module used in MM-Net and MMC-Net. The R@K results on Pascal Sentence are reported. By comparison, the convolutional fusion module can achieve better results than others.

Method	Fusion module	Image to Text			Text to Image		
		R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	No	45.0	82.0	91.0	35.6	75.8	87.0
MM-Net	Sum	46.0	83.0	91.0	36.8	77.6	87.6
MM-Net	Multiplication	46.0	84.0	91.0	37.2	78.4	87.6
MM-Net	Convolution	47.0	85.0	92.0	38.4	80.6	88.6
MMC-Net	No	51.0	85.0	92.0	37.6	80.6	92.0
MMC-Net	Sum	51.0	86.0	92.0	38.4	81.0	92.0
MMC-Net	Multiplication	51.0	86.0	92.0	39.0	81.0	92.0
MMC-Net	Convolution	52.0	87.0	93.0	41.0	81.2	92.5

Table 7.7: Analysis of the CBP module in MMC-Net. The R@K results on Pascal Sentence are reported, which demonstrate the effectiveness and efficiency of using the CBP module.

Method	Dimension	Image to Text			Text to Image		
		R@1	R@5	R@10	R@1	R@5	R@10
MMC-Net with FC	1024	50.0	86.0	92.0	39.6	80.4	90.0
MMC-Net with BP	262144	53.0	88.0	93.0	41.5	81.5	92.5
MMC-Net with CBP	2048	52.0	87.0	93.0	41.0	81.2	92.5

obtain considerable improvements over the one with FC. The MMC-Net with BP achieves better results than other methods, while its multi-modal representation has higher dimensionality. Instead, CBP can maintain both accuracy and efficiency.

Analysis of combining vision and language

This experiment is used to verify the advantage of incorporating visual and textual representations. As reported in Table 7.8, we compare the results between combining visual and textual features (*i.e.* MMC-Net) and using only visual features. We can observe that combining vision and language can achieve significantly superior accuracies on Flowers and CUB-Bird. Although visual features can enable the models to achieve promising performance, the informative textual features can further help improve the classification accuracies. This shows the effectiveness of capturing multi-modal representations from both vision and language. Furthermore, Figure 7.12 analyzes the test rates during the training iterations. It can be seen that the vision and language model can consistently outperform the vision model in the entire training stage.

Table 7.8: Analysis of combining vision and language. We report the Top-1 classification rates on Flowers and CUB-Bird. The model with both vision and language outperforms the model with only vision.

Method	Flowers	CUB-Bird
Only Vision	92.2	78.8
Vision and Language	95.2	82.4

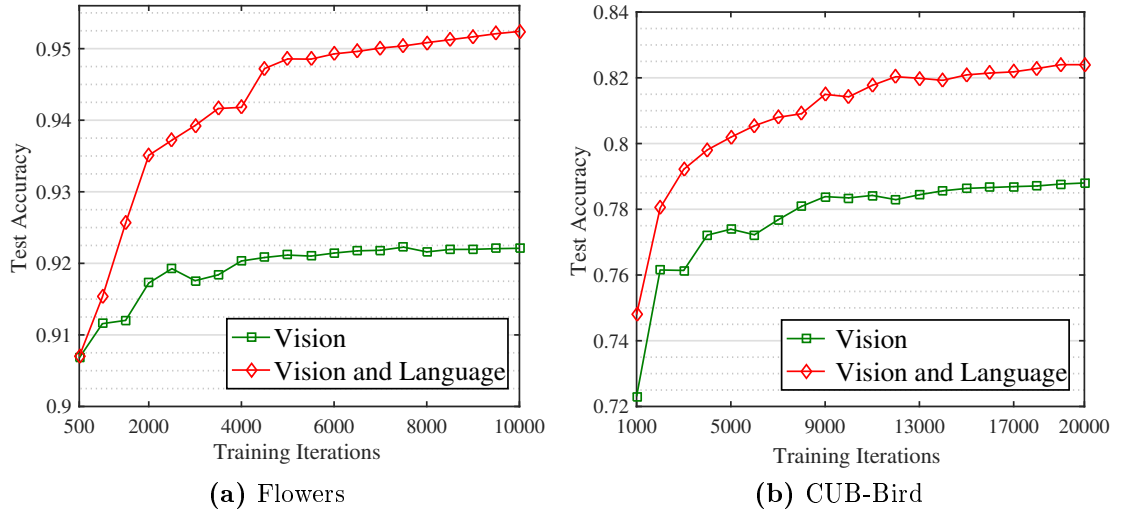


Figure 7.12: Illustration of the test classification rates during the training iterations. Incorporating language and vision is significant to improve the performance, compared to only using visual information.

Analysis of image encoders

As aforementioned in Section 7.2, we employ the ResNet-152 model to encode the input image. In this experiment, we aim to study the effect of different image encoders. For a fair comparison with DSPE [53], we provide the results of MMC-Net with VGG-19. Also, we implement the DSPE with ResNet-152. Table 7.9 reports the compared results on MSCOCO. For both VGG-19 and ResNet-152, our MMC-Net can outperform DSPE across all the measurements. We should realize that the improvements of MMC-Net come from two aspects. First, the matching component in MMC-Net has more layers than that of DSPE, *i.e.* four layers *v.s.* two layers. Second, MMC-Net utilizes a classification component to help improve the matching performance. This is the main motivation in this work. Note that, both MMC-Net and DSPE in Table 7.9 use the Mean vector to encode the input text. In [53], they also present another expensive textual representation using the Hybrid Gaussian-Laplacian mixture model (HGLMM) [51], *i.e.* a 18000-dimension vector. Currently, we do not introduce HGLMM to MMC-Net, even though it can help increase the performance.

Table 7.9: Analysis of image encoders. The image feature dimensions are also presented. MMC-Net has better matching results on MSCOCO than DSPE [53].

Method	Image encoder	Dimension	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
DSPE	VGG-19	4096	40.7	74.2	85.3	33.5	68.7	83.2
MMC-Net	VGG-19	4096	46.0	79.7	89.2	38.9	73.5	87.5
DSPE	ResNet-152	2048	53.1	82.7	90.2	43.5	78.2	88.9
MMC-Net	ResNet-152	2048	57.0	85.8	92.7	46.2	80.8	90.5

Table 7.10: Comparison with other state-of-the-art approaches on the Pascal Sentence dataset for image-text retrieval. Best results are in bold face.

Method	Image encoder	Text encoder	Image to Text		Text to Image	
			R@1	R@5	R@1	R@5
SDT-RNN [220]	AlexNet	DT-RNN	23.0	45.0	16.4	46.6
kCCA [220]	AlexNet	word2vec	21.0	47.0	16.4	41.4
DeViSE [214]	AlexNet	skip-gram	17.0	57.0	21.6	54.6
SDT-RNN [220]	RCNN	DT-RNN	25.0	56.0	25.4	65.2
DFE [181]	RCNN	word2vec	39.0	68.0	23.6	65.2
Mean Vector [51]	VGG-16	word2vec	52.5	83.2	44.9	84.9
GMM+HGLMM [51]	VGG-16	HGLMM	55.9	86.2	44.0	85.6
Proposed MMC-Net	ResNet-152	word2vec	52.0	87.0	41.0	81.2

7.4.6 Comparison with other approaches

For Pascal Sentence and MSCOCO, we compare our matching results with other state-of-the-art approaches. As reported in Table 7.10 and 7.11, MMC-Net can achieve competitive performance with the state-of-the-art. To be more specific, the method in [51] is effective on small-scale datasets, so it can obtain state-of-the-art results on Pascal Sentence. However, it does not have a strong generalization on large-scale datasets, for example their results on MSCOCO are not quite competitive. In contrast, the proposed MMC-Net maintains the high performance on both of small-scale and large-scale datasets. Moreover, we show the image and text encoders used in different approaches. Both of DSPE [53] and 2WayNet [76] extracted the visual features based on the VGG-19 model, while they rely on a more complicated HGLMM textual representation [51] than the Mean vector used in MMC-Net. As discussed earlier (Section 7.2), we did not use the HGLMM representation in order to maintain the training efficiency. For a fair comparison, MMC-Net with VGG-19 and Mean vector (see Table 7.9) can outperform DSPE with significant improvements, and can compete with 2WayNet while it uses the HGLMM representation. Lastly, we clarify that any common feature encoders for images and texts can be potentially adopted to MMC-Net. Exploring more efficient feature encoders is a fundamental and promising work.

For Flowers and CUB-Bird, we compare the fine-grained classification results with the state-of-the-art. Table 7.12 reports the comparison details. Since the compared methods do not utilize textual representations, we instead show the CNN model

Table 7.11: Comparison with other state-of-the-art approaches on the MSCOCO dataset for image-text retrieval. Best results are in bold face.

Method	Image encoder	Text encoder	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
DVSA [55]	RCNN	RNN	38.4	69.9	80.5	27.4	60.2	74.8
Mean vector [51]	VGG-16	word2vec	33.2	61.8	75.1	24.2	56.4	72.4
GMM+HGLMM [51]	VGG-16	HGLMM	39.4	67.9	80.9	25.1	59.8	76.6
m-RNN [190]	VGG-16	RNN	41.0	73.0	83.5	29.0	42.2	77.0
RNN-FV [185]	VGG-19	RNN	41.5	72.0	82.9	29.2	64.7	80.4
mCNN(ensemble) [52]	VGG-19	CNN	42.8	73.1	84.1	32.6	68.6	82.8
DSPE [53]	VGG-19	word2vec	40.7	74.2	85.3	33.5	68.7	83.2
DSPE [53]	VGG-19	HGLMM	50.1	79.7	89.2	39.6	75.2	86.9
2WayNet [76]	VGG-16	HGLMM	55.8	75.2	-	39.7	63.3	-
Proposed MMC-Net	ResNet-152	word2vec	57.0	85.8	92.7	46.2	80.8	90.5

Table 7.12: Comparison with other approaches on the Flowers and CUB-Bird datasets. Best results are in bold face. The methods in the upper part fine-tune the original CNN models, however, the ones in the lower part do not perform the fine-tuning process. We do not use the bounding box annotations in the datasets. Note that, we use the numbers to describe the depth of the image encoders. The dimension of MMC-Net indicates the multi-modal representation extracted from CBP.

Method	Image encoder	Finetune	Dimension	Flowers	CUB-Bird
Deep Optimized [224]	CNN-16	Yes	4096	91.3	67.1
Part R-CNN [225]	DeCAF-8	Yes	4096	-	76.5
Two-level attention [226]	AlexNet-8	Yes	4096	-	77.9
Deep LAC [227]	AlexNet-8	Yes	12288	-	80.3
NAC-const [221]	AlexNet-8	Yes	4096	91.7	68.5
NAC-const [221]	VGG-19	Yes	4096	95.3	81.0
Bilinear CNN [222]	VGG-16	Yes	250k	-	84.0
PD+FC+SWFV-CNN [223]	VGG-16	Yes	70k	-	84.5
MsML+ [228]	DeCAF-8	No	134016	89.5	67.9
BoSP [229]	VGG-16	No	5120	94.0	-
RI-Deep [230]	VGG-19	No	4096	94.0	72.6
ProCRC [231]	VGG-19	No	5120	94.8	78.3
MG-CNN [232]	VGG-19	No	12288	-	81.7
Proposed MMC-Net	ResNet-152	No	4096	95.2	82.4

used in the image encoder and the network depth. Note that, these approaches are divided into two groups based on whether the CNN model is fine-tuned on the target dataset. First, it can be seen that, MMC-Net achieves better results than other approaches without performing the fine-tuning step. Second, MMC-Net can even compete with the approaches with the fine-tuning step. For example, our results on Flowers is competitive with NAC-const [221]. Also, our approach is superior over most approaches on CUB-Bird, except Bilinear CNN [222] and PD+FC+SWFV-CNN [223]. However, we can see that both [222] and [223] produce a significantly more expensive feature vector than MMC-Net. We should realize that additional fine-tuning techniques have potential to improve performance, but are not the focus of this work. Our competitive results are partly due to the use of the ResNet-152 model, while we believe this should not decrease the effectiveness of our approach.

Table 7.13: Summary of the parameters used in the MMC-Net for matching and classification, and the time for running the multi-stage training algorithm.

Dataset	#Params for matching	#Params for classification	Time (hours)
Pascal Sentence	~8 millions	~41,000	~0.3
MSCOCO	~8 millions	~164,000	~7.0
Flowers	~8 millions	~418,000	~0.5
CUB-Bird	~8 millions	~820,000	~1.3

7.4.7 Computational cost

We conducted the experiments on a NVIDIA TITAN X card with 12 GB memory. In practice, we first extracted visual and textual features for all training samples using the off-the-shelf feature encoders. Then, we take as input these input features for the matching and classification components. Since the network parameters in MMC-Net are not expensive, it is feasible and rewarding to use a large mini-batch size to improve the training. In Table 7.13, we show the training parameters in the matching and classification component, and the multi-stage training time cost on the four datasets. The MSCOCO dataset consumes more training time due to its large-scale data. In summary, MMC-Net is an efficient network with a decent model complexity.

7.5 Chapter Conclusions

In this work, we proposed a unified network for joint multi-modal matching and classification. The proposed MMC-Net could simultaneously learn latent embeddings in the matching component, and generate a multi-modal representation vector in the classification component. Consequently, the two components could help promote each other by combining their loss functions together. We evaluated our approach on four well-known multi-modal datasets. The experimental results demonstrated the robustness and effectiveness of the MMC-Net model, compared to the baseline models. In addition, our approach achieved competitive results with the state-of-the-art approaches. The results showed its promising generalization for diverse multi-modal tasks related to matching or classification.

Future work. Currently, we use the class labels to train the classification component in MMC-Net. One potential improvement is to use more detailed information to guide the classification, like attributes. Compared to the class labels, attributes can discover more clues (*e.g.* sit, run, blue and small) about the visual content and text description. Hence, using attributes is beneficial for narrowing the gap between visual features and language words.