Large scale visual search

Wu, S.

Large Scale Visual Search

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 22 december 2016
klokke 16.15 uur

door

Song Wu

geboren te Sichuan, China
in 1985

Promotiecommissie

Promotor:	Prof. Dr. J.N. Kok	
Co-promotor:	Dr. M.S. Lew	
Overige leden:	Prof. Dr. A. Plaat	
	Prof. Dr. W. Kraaij	
	Prof. Dr. T.H.W. Bäck	
	Prof. Dr. C. Griwodz	(University of Oslo)
	Prof. Dr. M. Larson	(Delft University of Technology)

ISBN/AEN 9789463321174

Cover photo: The cover photo shows the flowchart of the proposed deep binary codes used for large scale visual search.

This research is financially supported by the China Scholarship Council (CSC), Grant No. 201206990003.

Contents

CONTENTS

Chapter 1

Introduction

Content-based image retrieval (CBIR) is one of the important and challenging problems in computer vision research. Frequently used search engines like Google or Yahoo represent a category of text-based information retrieval. However, the search accuracy of text-based information retrieval cannot satisfy the requirements of users when the text annotations are incomplete or incorrect. Moreover, due to the non-scalability of text-based information retrieval to large scale datasets, especially for the ever increasing multimedia data on the web, a high degree of manual effort is required to define the correct text annotations. Therefore, research on content-based information retrieval was proposed to address this issue.

CBIR aims to obtain more images related to the query by analyzing and exploring the content in images. The classic features (e.g., color, shape, texture, and etc.) are low-level features, which do not easily translate to the high level human concept vocabulary. In order to bridge the semantic gap, Bag-of-Words (BoW) [1] with local features was proposed to represent images and demonstrated high performance in image retrieval. Inspired by the BoW model, Fisher Vector (FV) [2], Vector of Locally Aggregated Descriptors (VLAD) [3], as well as their variants [4, 5, 6, 7] were proposed to seek a more informative image representation. Additionally, hash techniques were designed to map real valued image representations to binary codes such that large scale image search can be carried out in

an efficient way.

Benefitting from the advantages of convolutional neural networks (CNNs) [8] trained on sufficiently large and diverse datasets such as ImageNet [9], image representations based on the activations within CNNs have shown significant high performance regarding the state-of-the-art of various computer vision applications. Recent work focuses on investigating effective ways to aggregate the activations within CNNs into compact and distinctive image representations for large scale image search.

Our main research objectives in this thesis are as follows:

- design a robust local descriptor to improve the discrimination of BoW based image representations.
- provide an effective way for large scale image search by exploring compact deep binary codes from deep layers within CNNs.
- propose a more powerful CNN architecture to improve the robustness of the image representation generated from the deep layers in CNNs.

1.1 Salient Point Methods

Generally, salient point methods consist of two parts: a local detector and a local descriptor. A robust local detector should have high repeatability. For compared images with the same object or scene, a high percentage of corresponding salient points should be detected on the scene part visible in both images. At the same time, they should be unaffected by various deformations, such as image blur, affine deformations, compression, and noise. Distinctive and robust local descriptors should be invariant and less sensitive to different deformations, such that they can be distinguished and effectively matched. Most of the existing salient point methods are invariant to transformations of scale, rotation, viewpoint and noise. All these properties are achieved by three distinct steps in salient point methods: scale space construction, orientation assignment and local descriptor generation.

Scale space construction: the construction of scale space aims to solve the challenges in computer vision where the vision information is captured at different scales. The commonly used scale space is the Gaussian smoothed image pyramid which can represent an image at multi-scales and multi-resolutions. The multi-scale representation is achieved by convolving the original image with different Gaussian smooth kernels, and the multi-resolution representation is achieved by down sampling or spatial pooling of the original image.

The experimental evaluation in the work by Lindeberg [10] demonstrated that the extreme points detected in a scale space constructed by scale-normalized Laplace-of-Gaussian (LoG) show high stability and repeatability. The representation of LoG scale space is constructed by smoothing the high resolution image with derivatives of Gaussian kernels of increasing size. Recent work focuses on the efficiency of scale space construction. An efficient framework of Difference-of-Gaussian (DoG) [11], which is an approximation of the Laplace-of-Gaussian (LoG) was proposed. A Difference-of-Gaussians pyramid is computed from the differences between the adjacent levels in the Gaussian pyramid. Then, the points at which the DoG values assume extrema of the differences with respect to both the spatial coordinates in the image domain and the scale level in the pyramid can be considered as candidate salient points. Compared to LoG, the DoG operation could significantly decrease the computational complexity. As the second order Gaussian derivatives (Hessian matrix) can also be used to approximate the LoG and the Hessian matrix can be further approximated by a box-filter, the box-filter based scale space [12] makes use of the box-filter and integral images to calculate the determinant of the Hessian matrix at a very low computational cost. The Gaussian smoothing operation in LoG, DoG as well as box-filter based scale space construction is linear and each pixel in the Gaussian image pyramid layer is convolved with the same Gaussian kernel. Hence, it can cause blurring as well as loss of the boundary details of the object in images. The introduction of nonlinear scale space can reduce noise but retain the object boundary structure such that we can obtain accurate positions of extreme points. The framework of nonlinear scale space first convolves the image with a Gaussian kernel of standard deviation. Then it builds the nonlinear scale space in an iterative way via

the additive operator splitting (AOS) scheme [13]. Since the extreme points are detected in discrete scale space, the final scale and location information of these extreme points can be refined by a quadratic fitting operation.

Orientation assignment: assigning each detected salient point an orientation. The local descriptor of the salient point in images can be represented relative to the calculated orientation and therefore be invariant to image rotation. The orientation assigned by a good measurement can make the generated local descriptor more robust to even large image rotation changes. The general orientation measurements can be categorized as: histogram of gradients [14], Haar-wavelet response [12], intensity centroid [15], average gradient of sampling pairs in a pre-designed pattern [16, 17, 18]. The histogram of gradient based orientation is calculated according to the gradient of each pixel within a patch around the salient point $\theta(x, y) = \arctan(\bigtriangledown I(x, y))$. Then, all $\theta(x, y)$ values are counted to generate a histogram and the maximum bin in the histogram is considered to be the orientation of the salient point. The Haar-wavelet response based orientation is computed according to the responses of Haar-wavelets in both horizontal and vertical directions in a circle region around the salient point. The dominant orientation of the local region is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of 60 degrees, and the largest response is considered as the orientation of the salient point. The intensity centroid approach assumes that the intensity of a salient point is an offset from its center, and it can be used to compute the moments of a patch and further to find its centroid. The orientation is defined as the direction of the vector from the salient point position to the centroid position in the local patch. Some other methods assign orientations to the salient points by averaging the sum of local gradients of the defined pairs in the pre-defined structure.

Local descriptor generation: the local descriptor generation is performed on image data that has been transformed relative to the assigned scale, location, and orientation such that they can be distinguished and matched under these transformations. The existing local descriptor can be categorized as a hand-crafted based scheme and a machine learning based scheme.

Hand-crafted local descriptors: the hand-crafted schemes mainly explore the intensity patterns around the detected salient points. The most representative descriptors are the distribution-based local descriptors, which include successful representations such as: histogram of gradients, histogram of gradient orientations, and Haar-wavelet responses distributions, which represent distinctive visual information according to the distributions of pixel intensities in the local patch. Binary local descriptors were proposed with an emphasis on minimizing computational and storage costs. The binary string representations make use of simple pair-wise pixel intensity comparisons. Different binary string local descriptors advocate different pre-defined structures to select the pixel pairs and the generated binary codes can be very efficiently matched with low computational cost using the Hamming metric (bitwise XOR followed by a bit count).

Machine learning based local descriptors: machine learning has been applied to improve both the efficiency and accuracy of local binary descriptor generation. Hashing is one of the most effective techniques which aims to construct a set of hash functions to map the original input space to compact and similarity-preserving local binary codes. Therefore, similar input spaces could be projected to similar binary codes in the Hamming space. Existing hashing approaches can be divided into two categories: data-independent and data-dependent methods. Data-independent methods randomly generate a projection matrix to map image features into binary representations without training data [19]. The representative methods are locality-sensitive hashing (LSH) [20] as well as its variants [21, 22]. Data-dependent hashing which is also referred to as a hashing learning approach focuses on learning hash functions from a specific training dataset.

Hash learning approaches involve two main steps. First, the training data is represented as hand-crafted real valued local features. Second, optimize an objective function to learn the hash function and use the learned hash function to convert the real valued input space representation into a binary representation. Generally, the process of hash functions learning is either done in an unsupervised, semi-supervised or supervised manner. Unsupervised hash function learning makes use of the unlabeled training data, and learns the compact binary descriptors whose Hamming distance is correlated with data similarity in the original input space

[23, 24, 25, 26, 27]. Supervised and semi-supervised hashing approaches take advantage of semantic label information of training data to preserve the ground truth similarity during the construction of binary hash codes. Supervised hashing fully exploits the labeled training data to seek a linear transformation. A loss function is usually defined that penalizes the reconstruction errors between the distances of original data and the distances of corresponding binary data to learn the hash functions [28, 29, 30, 31, 32]. For the case of semi-supervised hashing learning, both unlabeled data and labeled data are utilized to learn the hash functions. For example, the semi-supervised hashing frameworks proposed by Wang et al. [33, 34] minimize the empirical error on the labeled data while maximizing the variance over labeled and unlabeled data for binary representations. Recently, the deep supervised hashing methods were proposed to learn binary hashing codes. Deep supervised hashing trains a deep hierarchical and nonlinear transformation model and projects the original local descriptors into local binary codes [35, 36, 37, 38].

1.2 Visual Word based Image Search

Content-based image search is still a challenging problem in computer vision. This is mainly due to the existing variations in image appearance, such as the changes of scale, orientation, viewpoint and illumination. In addition, with the increasing amounts of image data on the web, a robust image representation with the approximate nearest neighbor (ANN) search has been widely used for large scale image retrieval. This method is mainly benefiting from the robustness of local descriptors to various geometric transformations and the applicability of different similarity measures.

The Bag-of-Words (BoW) model which is inspired from simple document retrieval systems and based on the analogy of visual words, has been widely applied in content-based image retrieval. In the BoW model, salient regions are first detected from each image in the training dataset and a high dimensional descriptor is calculated for each region. These descriptors are then clustered to

form a vocabulary of visual words. Therefore, an image is finally represented as a histogram over a set of learned visual words after quantizing each of the local descriptors to the nearest visual word. Early systems [1] used a flat K-means clustering to generate the visual vocabulary, but it was difficult to scale to large vocabularies generation and large scale datasets. The later works [39, 40] show that flat K-means can be scaled to similarly large vocabulary sizes by the use of approximate nearest neighbor methods.

The Fisher Vector (FV) [2] image representation seeks to capture the probability distribution of features. The generative model Gaussian mixture model (GMM) is utilized in FV to estimate a parametric probability distribution over the feature space from a large representative set of local descriptors. The local descriptors extracted from the image dataset are assumed to be sampled independently from this probability distribution. Each local descriptor is represented by the gradient of the probability distribution at that feature with respect to its parameters. Gradients corresponding to all the features with respect to a particular parameter are summed. The final FV representation is the concatenation of the accumulated gradients. They achieve a fixed length vector from a varying set of features that can be used in various discriminative learning activities. Compared with the K-means cluster algorithm, GMM delivers not only the mean information of visual words, but also the shape of their distribution.

Jegou et al. proposed Vector of Locally Aggregated Descriptors (VLAD) [3] which can be viewed as a simplified non-probabilistic version of Fisher Vector. Similar to the BoW model, a vocabulary with C visual words is first learned via K-means cluster. Each local descriptor is associated to its nearest visual word in the vocabulary. The idea of the VLAD representation is to accumulate the residuals belonging to each of the visual words. This characterizes the distribution of the vectors with respect to the center. A number of variants of VLAD have also been designed to enhance the image representation by considering vocabulary adaptation and intra-normalization [4], residual normalization and local coordinate systems [5], geometry information [6] and multiple vocabularies [7].

The relationship among the models of BoW, FV and VLAD can be described as: BOW encodes the 0-order statistics of the distribution of local descriptors, the

Fisher vector extends the BOW by encoding high-order statistics (first-order and, optionally, second-order), and VLAD is a non-probabilistic equivalent of Fisher Vector. During the past decade, the visual words based image representation has been successfully applied in various computer vision applications.

1.3 Convolutional Neural Networks

Generally, the convolutional neural network (CNN) is a hierarchical architecture [8] which consists of several stacked convolutional layers (optionally followed by a normalization layer and a spatial pooling layer), fully connected layers and a loss layer on top. The convolutional layers generate feature maps by linear convolutional filters followed by nonlinear activation functions (Rectifier, Sigmoid, TanH, etc.). The fully connected layer has full connections to all activations in the feature maps and the resulted one dimensional vector can be fed into the loss layer for loss function optimization.

There are two main stages for training the convolutional neural network: a forward stage and a backward stage. First, the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then the output from the last layer is used to compute the loss function with the ground truth labels. Second, based on the loss cost, the backward stage computes the gradients of each parameter with chain rules. All the parameters are updated based on the gradients, and are prepared for the next forward computation. After sufficient iterations of the forward and backward stages, the network could be optimized. The convolutional neural network has been applied in diverse computer vision applications and demonstrated their significant advantages and high performance.

We will first present an overview of the different types of layers and then briefly review the CNN based computer vision applications.

Convolutional layers: the convolutional layers in the CNN architecture utilizes k filters (or kernels) to convolve the input image to generate k feature maps. There are three main advantages of the convolution operation [41]: first, the

parameter sharing mechanism is used in convolutional layers such that the number of parameters could be significantly reduced. Second, local connectivity learns correlations among neighboring pixels. Third, it is invariant to the location of the object. Due to these benefits introduced by the convolution operation, some well-known research papers also use it as a replacement for the fully connected layers to accelerate the learning process [42, 43].

Pooling layers: a pooling layer is an optional layer following a convolutional layer which sub-samples its input. Average pooling and max pooling are the most commonly used pooling operations. The reason to use a pooling operation in the convolutional neural network is that: first, it can reduce the dimensions of the output and the number of parameters of the network, while keeping the most salient information. Second, a pooling operation also provides basic invariance to translating (shifting) and rotation. For max pooling and average pooling, Boureau et al. [44] provided a detailed theoretical analysis of their performances. Scherer et al. [45] further conducted a comparison between the two pooling operations and found that max-pooling can lead to faster convergence, selection of superior invariant features and improve generalization.

Fully-connected layers: after several convolutional and max pooling layers, the high-level reasoning in the convolutional neural network is done via the fully connected layers. A fully connected layer takes all neurons in the previous layer (be it fully connected, pooling, or convolutional) and connects it to every single neuron it has. Fully connected layers are not spatially located, as the input feature maps are converted to a one dimensional feature vector. The one dimensional feature vector could either feed forward the vector into a loss layer or take it as a feature representation for follow-up processing [46]. The drawback of the fully connected layer is that it contains many parameters, which results in large computational and storage costs. Therefore, GoogleNet [42] designed a deep and wide network while keeping the computational budget constant, by switching from fully connected to sparsely connected architectures. The Network in Network (NIN) [47] architecture replaces the fully connected layer by a global average pooling layer.

Recently, deep learning, especially for the CNNs, produced state-of-the-art performance on various computer vision applications, such as image classification, image search, object detection, semantic image segmentation, human pose estimation, etc.

Image classification: Krizhevsky et al. [48] set a milestone for large-scale image classification when training a large CNN on the ImageNet dataset [9], thus proving that CNN could perform well on natural image classification. OverFeat [49] proposed a multi-scale and sliding window framework, which could find the optimal scale of the image and fulfill different tasks simultaneously, e.g., image classification, object detection and localization. Because the existing CNNs require fixed-size image data as input, the SPP-Net [50] model eliminated this restriction via a spatial pyramid pooling strategy in the CNNs and improved the classification accuracy of a variety of CNN architectures despite their different designs. The later proposed VGGNet [51] and GoogleNet [42] significantly improved the performance of image classification by increasing the width and depth of the network architectures.

Object detection: a general scheme for high performance object detection systems is to generate a large number of candidate object region proposals and classify them using their high performance CNN features. The most representative approach is the regions with CNN features (RCNNs) [46]. It utilizes selective search [52] to generate object region proposals, and extracts the CNN features for each candidate region. The features are then fed into a SVM classifier to decide whether the related candidate windows contain the object or not. RCNNs improved the benchmark of object detection by a large margin, and became the base model for many other promising algorithms [53, 54, 55]. Also, the original RCNNs were computationally expensive, the recent works [50, 56] employed the strategy of sharing convolutions across the region proposals to reduce the computation cost. The latest frameworks of Fast RCNNs [56] and Faster RCNNs [57] achieve near real-time rates using very deep networks.

Image retrieval: The success of AlexNet [48] suggests that CNNs can be used as high level and universal feature extractors. The features emerging in the fully connected layers of the CNN can serve as a high level image representation

for image classification. Motivated by this, many recent studies make use of the activations of the top fully connected layers in CNNs for image retrieval [58, 59, 60, 61, 62, 63]. Recent researches also suggested to explore the features from the deep convolutional layers in CNNs. These features have very useful properties: first, they can be efficiently extracted from an image of any size and aspect ratio. Second, features from the convolutional layers have a natural interpretation as descriptors of local image regions corresponding to receptive fields of the particular features. Finally, simple pooling operations can aggregate feature maps from deep convolutional layers into low dimensional and highly distinctive features [58, 60, 63, 64, 65]. The CNNs based image representations have demonstrated their competitive and even better results compared with the traditional visual words methods, such as BoW, VLAD and Fisher Vector.

Semantic image segmentation: semantic image segmentation can be referred to as a problem of pixel-level classification or labeling. Recently, the CNNs and probabilistic graphical models were utilized to address this task and yielded promising progress [66, 67, 68, 69, 70]. The CNN based semantic image segmentation methods usually convert an existing CNN architecture constructed for classification to a fully convolutional network (FCN). This is mainly because the FCN architecture accepts a whole image as an input and performs fast and accurate inference. Long et al. [68] replaced the last fully connected layers of a CNN by convolutional layers, and obtained a coarse label map from the network by classifying every local region in the image, then performed a simple deconvolution, which is implemented as a bilinear interpolation, for pixel-level labeling. DeepLab [69] proposed a similar FCN based model which obtained denser score maps within the FCN framework to predict pixel-wise labels and refined the label map using the fully connected conditional random field (CRF). Instead of using CRF as a post-processing step, Lin et al. [70] jointly trained the FCN and CRFs by efficient piece wise training.

Human pose estimation: estimating the human pose by locating human body joints or facial landmarks is a challenging task, because of the changes in pose, illumination, occlusion and etc. As CNNs have shown outstanding performance on visual classification and object localization, human pose estimation can be

formulated as a CNN-based regression problem towards human body joints. The representative projects [71, 72] proposed to use a cascade of CNN-based regressors to reason the positions of body joints or facial landmarks. The cascade of CNNs can be viewed as a kind of holistic process which takes the full image as the input and output the ultimate position of body joints or facial landmarks in the image without using any explicit graphical model or part detectors. The part-based processing methods propose to detect the human body parts individually, followed with a graphical model to incorporate the spatial information. Rather than training the network using the whole image as input, Chen et al. [73] utilized the local part patches to train a CNN, in order to learn conditional probabilities of the part presence and spatial relationships. By incorporating with graphical models, the algorithm gained promising performance. Tompson et al. [74] designed multi-resolution ConvNet architectures to perform heat-map likelihood regression for each body part, followed with an implicit graphical model to further promote joint consistency. The model was further extended and improved [75], which argues that the pooling layers in the CNN would limit spatial localization accuracy and try to recover the precision loss caused by the pooling process. Additionally, Fan et al. [76] proposed a dual-source convolutional neutral network (DS-CNN) to integrate the holistic and partial view in a two branche CNN architecture. It takes part-patches and body-patches as inputs to combine both local and contextual information for more accurate pose estimation.

1.4 Thesis Overview

This thesis is based on first-authored conference papers that have been published or journal papers are currently under review. The research has been carried out during the four-year period of the PhD research. The focus of this thesis has been on developing and analyzing techniques to improve the state-of-the-art of large scale image search.

Chapter 2: A Comprehensive Evaluation of Salient Point Methods

A survey is presented that evaluates the performance of a wide set of salient point detectors and descriptors. First, the survey compares diverse salient point algorithms with regard to the repeatability of salient point detectors, recall and precision of salient point descriptors. Then, it integrates the salient point methods with the framework of fully affine space and evaluates the performance under major viewpoint transformations. The presented comparative experimental results can benefit researchers in choosing an appropriate detector and descriptor for different computer vision applications. An early version of this work was presented at:

- 21st ACM international conference on Multimedia (MM 2013), in Barcelona, Spain.

Chapter 3: RIFF: Retina-inspired Invariant Fast Feature Descriptor

We first propose the Retina-inspired Invariant Fast Feature, RIFF, which was designed for invariance to scale, rotation, and affine image deformations. The RIFF descriptor is based on pair-wise comparisons over a sampling pattern loosely based on the human retina and introduces a method for improving accuracy by maximizing the discriminatory power of the point set. A performance evaluation with regard to Bag-of-Words based image retrieval on several well-known benchmark datasets demonstrates that the RIFF local descriptor has competitive performance to the state-of-the-art local descriptors. Additionally, a popular approach from the literature is to use visual words (or Bag-of-Words) constructed from real valued descriptors (SIFT and SURF). To accommodate large scale data sets, we used the approximate nearest neighbor (ANN) based cluster approach to both real valued local descriptors and binary string local descriptors (BRIEF, ORB, BRISK and FREAK) and the results on the test datasets reveal that some of the recent binary string approaches outperformed notable descriptors such as SIFT and SURF. This approach has been presented at the following conferences:

- 22nd ACM international conference on Multimedia (MM 2014), in Orlando, FL, USA.
- 4th ACM International Conference on Multimedia Retrieval (ICMR 2014), in Glasgow, Scotland.

Chapter 4: Deep Binary Codes for Large Scale Image Retrieval

We present a novel and effective method to create compact binary codes (deep binary codes) based on deep convolutional features for image retrieval. Deep binary codes are generated by comparing the response from each feature map and the average response across all the feature maps on the deep convolutional layer. Additionally, a spatial cross-summing strategy is proposed to directly generate bit-scalable binary codes. As the deep binary codes on different deep layers can be obtained by passing the image through the CNN and each of them makes a different contribution to the search accuracy, we then present a dynamic, on-the-fly late fusion approach where the top N high quality search scores from deep binary codes are automatically determined online and fused to further enhance the retrieval precision. Two strengths of the proposed methods are that the generation of deep binary codes is based on a generic model which does not require additional training for new domain areas, and the dynamic late fusion scheme is query adaptive. Extensive experimental results on well known benchmarks show that the deep binary codes are competitive to state-of-the-art approaches in terms of the performance on large scale image retrieval. Moreover, the search accuracy is shown to be enhanced substantially by the dynamic late fusion scheme. The paper has been submitted to:

- Journal of Neurocomputing

Chapter 5: Comparison of Information Loss Architectures in CNNs

We propose a novel deep convolutional neural network called "Weighted Integration Architecture Network" (WIAN) which can effectively recover the information loss due to the pooling operation in the CNNs. The proposed WIAN reuses the information from the previous layers in the network and assigns a weight matrix to each layer and then integrates them via an element-wise sum operation to further enhance the performance of image classification. Several types of weight scheme such as adaptive weight learning framework as well as responses or entropy based weight learning schemes have been evaluated in this chapter. Extensive experiments on four standard benchmark datasets demonstrate the effectiveness as well

as an improved performance of the proposed WIAN. The basis for this chapter is formed by the publication in the conference proceeding:

- 17th Pacific-Rim Conference on Multimedia (PCM, 2016) in Xi'an China.

These are the publications which were related to the contents of this thesis:

- **Wu S.**, and Lew M.S., "Evaluation of salient point methods." Proceedings of the 21st ACM International Conference on Multimedia. ACM, 2013.
- **Wu S.**, and Lew M.S., "Salient features for visual word based image copy detection." Proceedings of International Conference on Multimedia Retrieval. ACM, 2014.
- **Wu S.**, and Lew M.S., "RIFF: Retina-inspired Invariant Fast Feature Descriptor." Proceedings of the 22nd ACM International Conference on Multimedia. ACM, 2014.
- **Wu S.**, and Lew M.S., "Comparison of Information Loss Architectures in CNNs." Proceedings Pacific RIM Conference on Multimedia, 2016.
- **Wu S.**, and Lew M.S., "Image Correspondences Matching Using Multiple Features Fusion." Proceedings of European Conference on Computer Vision Workshop, 2016.
- **Wu S.**, Oerlemans A, Bakker E.M., and Lew M.S., "Deep Binary Codes for Large Scale Image Retrieval." Journal of Neurocomputing (submitted).
- **Wu S.**, Oerlemans A, Bakker E.M., and Lew M.S., "A Comprehensive Evaluation of Salient Point Methods." Journal of Computer Vision and Image Understanding (submitted).
- Guo Y., Liu Y., Oerlemans A., Lao S., **Wu S.**, and Lew M.S. "Deep learning for visual understanding: A review." Journal of Neurocomputing, vol 187, 2016.
- Zhang Q., Zaaijer S., **Wu S.**, and Lew M.S. "3D Image Browsing: The Planets". Proceedings of International Conference on Multimedia Retrieval. ACM, 2014.

- Guo Y., Bai L, Lao S., **Wu S.**, and Lew M.S. " A Comparison between Artificial Neural Network and Cascade-Correlation Neural Network in Concept Classification." Proceedings Pacific RIM Conference on Multimedia, 2014.
- Liu Y., Guo Y., **Wu S.**, and Lew M.S. (2015), "DeepIndex for Accurate and Efficient Image Retrieval." Proceedings of International Conference on Multimedia Retrieval. ACM 2015.

Chapter 2

A Comprehensive Evaluation of Salient Point Methods

As salient point methods can represent distinctive and affine invariant points in an image, various types of salient point methods have been proposed over the past decade. Each method has particular advantages and limitations and may be appropriate in different contexts. In this chapter, we evaluate the performance of a wide set of salient point detectors and descriptors. First, we compare diverse salient point methods with regard to the repeatability of detectors, and the recall and precision of descriptors. Next, we integrate the salient point methods with the framework of fully affine space and evaluate their performance under major viewpoint transformations. The presented comparative experimental studies can support researchers in choosing an appropriate detector and descriptor for their specific computer vision applications.

2.1 Introduction

Salient point methods which can describe meaningful, stable, and representative local features in an image have become prevalent in diverse areas in computer vision, such as object and scene recognition [77, 78], 3D object reconstruction [79], visual tracking [80, 81] and multimedia information retrieval [3, 18, 82, 83, 84, 85, 86, 87, 88]. Most of the salient point algorithms contain two parts: a detector and a descriptor. The detector locates a set of distinctive points which can be invariant to various transformations (e.g., scaling, translation, viewpoint changes), and the descriptor encodes the important information from the local patch centered on the salient point into a feature vector, which makes it possible to reliably match correspondences across different transformations of the same object or the same scene.

Typically, object recognition, 3D reconstruction and visual tracking mainly rely on the correctly matched correspondences between two compared images. These applications start by extracting local descriptors from each image and insert the obtained local descriptors into an index space for efficient correspondence matching. The RANSAC algorithm [89] is further adopted to eliminate outlier matches and to estimate the homography between the compared images. Therefore, a salient point detector with high repeatability and a local descriptor with discriminatory power is required for these applications.

However, accurate correspondence matching under large viewpoint changes is still a major challenge, because greater image viewpoint transformations result in a significant decrease of saliency and repeatability of salient points. Yu et al. [90] proposed to use the framework of fully affine space to overcome this issue. The basic idea behind the framework of fully affine space is that the projective transformation induced by camera motion around a smooth surface can be approximated by an affine transformation. A notable method is ASIFT which generates all image views in the whole affine space and extracts SIFT local features in these synthetic images to increase the matching precision. As the high dimensionality of the SIFT descriptor leads to a high computational complexity in the framework of fully affine space, we combine the recent lower computational

complexity salient point algorithms with the framework of fully affine space and evaluate their performance under the extreme viewpoint changes.

This chapter is an extension of our previous projects [87, 88] which provide a comparison guide of recently proposed salient point detectors and descriptors. The main contributions of this chapter are summarized as follows:

First, the repeatability performance and the computational cost of each salient point detector are presented.

Second, the efficiency and accuracy of both the real valued descriptors and binary string descriptors in terms of recall and precision on two benchmark datasets are evaluated.

Third, we calculate the accuracy and time complexity of each salient point method in the framework of fully affine space such that researchers could make a trade-off between precision and efficiency under extreme viewpoint changes.

2.2 Background

Early research on salient point methods mainly focused on finding high variance or corner points in the image. One of the first detectors was developed by Moravec [91] and it is defined according to the average intensity changes in different directions within the local region around a point. The Harris corner detector [92] defines a corner structure point, if its second-moment matrix has two large eigenvalues. The similar Hessian corner detector [93] determines a corner point in the image, if it is the local extrema of the Hessian matrix determinant. As both the Harris and Hessian detectors find the corner points at a fixed scale, the Harris-Laplacian and Hessian-Laplacian [94, 95] are designed to be scale invariant. Harris-Laplacian and Hessian-Laplacian locate corner candidates on each level of the scale space. Those points for which the Laplacian simultaneously attains local extrema over scales are selected as corner points. The FAST [96] detector identifies the corner points according to the criterion whether a set of

contiguous pixels in a circle are all brighter or all darker than the intensity of the centre point.

Since conventional corner point detectors are only invariant to scale, translation, and noise, affine covariant region detectors were developed to reduce the influence of viewpoint changes. The Harris-Affine detector and the Hessian-Affine detector [97] find the initial candidate points by using the Harris-Laplacian corner detector and Hessian-Laplacian corner detector, respectively, and then fit an elliptical region to each point via the second moment matrix of the intensity gradient. MSER [98] computes the connected binary regions through a large set of multiple thresholds, and the selected regions are those that maintain unchanged shapes over these thresholds. As edges are typically rather stable structures that can be detected over a range of image changes, EBR [99] starts by detecting corner points in an image and identifies the affine covariant region of each point by exploiting the edge information present nearby. IBR [100] detects intensity extrema at multiple scales and captures the intensity pattern along rays emanating from each extremum to define a region of arbitrary shape. The region of IBR is delineated by the image points defined over these rays where the intensity suddenly increases or decreases, and then uses an ellipse to fit the region. However, the operation of elliptical region fitting in the affine covariant detector could result in partial information loss.

Recent salient point methods focus on the repeatability and precision of the detector, as well as the distinctiveness, computational efficiency and low memory requirement of the local descriptor. The most representative one is SIFT, which efficiently builds the scale space by employing the Difference of Gaussians to approximate the Laplacian of Gaussians and represents the local descriptor using a gradient orientation histogram. Meanwhile, some variants of SIFT are proposed with the aim to increase the discrimination of the SIFT descriptor. PCA-SIFT [101] utilizes PCA to reduce the dimension of the original SIFT descriptor to further speed up the process of local descriptor matching. Color-SIFT [102] takes the color gradients, rather than intensity gradients in the local region around the salient point to generate the feature. Rank-SIFT [103] adopts a data-driven approach to learn a ranking function to sort the salient points such that the

unstable points can be discarded. Root-SIFT [104] adds a square root operation to the normalized SIFT features and uses the Hellinger kernel to increase the matching accuracy. DSP-SIFT [105] generates the descriptor through pooling the gradient histogram across different domain sizes of each salient point into a feature and it even outperforms the high level convolutional neural network feature [48]. Affine-SIFT (ASIFT) [90] is proposed with the aim to be perspective invariant and it does this by simulating images under various views to cover the whole affine space and extracting SIFT descriptors in all these simulated images for matching. Different from these variants of SIFT, other approaches target on improving the efficiency of scale space establishment or accuracy of salient points localization. For example, the SURF detector makes use of a box-filter and the integral image to speed up the scale space building. The ORB and BRISK detectors use a Gaussian image pyramid to efficiently establish the scale space. As the construction of scale space by linear multi-scale Gaussian pyramids easily results in the blurring and the loss of boundary details, KAZE [106] combines a nonlinear scale space with additive operator splitting (AOS) and special conductance diffusion to reduce noise while retaining the object boundary structure. The advantage of the nonlinear scale space in KAZE is that it could provide more accurate positions for salient points.

In order to meet the requirements of real time systems and devices with limited computational and storage resources, binary string local descriptors were recently introduced. Binary string representations make use of a pixel-pair intensity comparison to generate the binary code. The resulting binary code holds some significant advantages: first, the operation of intensity comparison is fast, the memory requirement of binary codes is low and matching binary codes via the Hamming distance is much faster than the Euclidean metric. A representative descriptor is BRIEF, which randomly samples a set of pixel-pairs from a Gaussian distribution in the smoothed local patch around the salient point and produces a binary string descriptor via the intensity comparison of pixel-pairs. The ORB descriptor integrates rotation invariance into BRIEF by estimating the orientation via the intensity centroid method. Additionally, ORB makes use of an unsupervised learning scheme to select pixel-pairs, rather than the random sam-

pling of BRIEF. BRISK and FREAK generate the binary string descriptors by comparing pair-wise intensities over a pre-defined pattern, a concentric ring-based sampling pattern and a retina sampling pattern, respectively. In contrast to those hand-crafted patterns, learning based approaches are proposed with the goal of closing the performance gap with real valued representations while maintaining the benefits of binary representations. BinBoost learns a set of hash functions using boosting and projects the image patch into a binary representation. LATCH proposes to learn patch triplet arrangements in the image and compares the intensity of triplet patches rather than the intensity of pixel-pairs to generate the binary codes.

Several related reviews present the performance evaluation of various salient point methods. Schmid et al. [107] uses the measure of "repeatability rate" and "information content" to evaluate the performance of different salient point detectors. Mikolajczyk et al. [108] made a performance evaluation of local descriptors by measuring the accuracy of matching and recognition. Accuracy and computational efficiency trade-offs [109] have been studied where different indexing structures were employed (such as approximate KD-trees). Heinly et al. [110] and Figat et al. [111] investigate the recall and precision of recent binary string representations under different image deformations. Gauglitz et al. [81] presents a comparison of different salient point methods on video object tracking. Moreels and Perona [112] made a performance evaluation of both feature detectors and descriptors on 3D object matching. Mukherjee et al. [113] made a performance evaluation for each combination of recent detectors and descriptors on object matching. To our knowledge, our review is the first one that evaluates the viewpoint invariance of each salient point approach in the fully affine space.

2.3 Overview of Evaluated Salient Point Methods

The aim of salient point methods is to extract distinctive invariant features from images that can be used to perform image correspondence matching and to per-

form the image representation. Recent salient point methods consist of four main procedures: the first step is to establish the scale space and find the extrema across all scales to achieve scale invariance. The second step is to determine the locations of the extrema and to define a local region for each according to the scale information. Then, each defined region is normalized and assigned a domain orientation to be rotation invariant. Finally, the region content is rotated based on the calculated orientation, after which, the discriminative information in the rotated region is encoded into a local descriptor. The existing schemes of local descriptor generation can be categorized into hand-crafted schemes and automatically learned schemes. The recent literature focuses more on the automatic learning of local descriptors. The learning based schemes usually optimize an objective function to generate robust and distinctive local descriptors. In particular, the most common objective functions are designed to minimize the distance between the descriptors from the same 3D coordinate (scale and location) or same class label extracted under varying imaging conditions and different viewpoints, meanwhile, maximizing that distance between patches from different 3D coordinates or different class labels. Table 2.1 gives an overview of all the evaluated salient points approaches in the experiments section.

2.3.1 SIFT (detector/descriptor)

SIFT proposed by Lowe [14] is the most popular salient point approach. The implementation of SIFT begins by building the Gaussian scale space which approximates the Laplacian-of-Gaussian function by the computationally efficient Difference-of-Gaussian function. It searches extrema over all scales to identify the potential salient points. Since the extreme points are detected in discrete scale space, it then uses the derivative of the Taylor expansion of the DoG function to determine the accurate scale and location for each salient point and simultaneously rejecting unstable extrema with low contrast. Furthermore, because a poorly defined extremum in the DoG function has a large principal curvature across the edge but a small one in the perpendicular direction, a Hessian matrix is employed to compute the principal curvatures and to eliminate points which are

Table 2.1: Overview of the evaluated salient point approaches in this chapter by Detector (Det.), Descriptor (Desc.), Scale Space, Orientation, and Descriptor Generation.

Methods	Det./Desc.	Scale Space	Orientation	Descriptor Generation
SIFT	yes/yes	Difference of Gaussian	local gradient histogram	local gradient histogram
SURF	yes/yes	box-filter	local Haar-wavelet responses	local Haar-wavelet responses
MSER	yes/no	no	no	no
HESSIAN-AFFINE	yes/no	no	no	no
FAST	yes/no	no	no	no
CenSurE	yes/no	bi-level filter	no	no
GFTT	yes/no	no	no	no
KAZE	yes/no	nonlinear scale space	no	no
BRIEF	no/yes	no	no	intensity comparison of pair-wise pixels in the random sampling pattern
ORB	yes/yes	Gaussian image pyramid	intensity centroid calculation	oriented BRIEF descriptor
BRISK	yes/yes	Gaussian image pyramid	average of the sum of the local gradient	intensity comparison of pair-wise pixels in concentric circles pattern
FREAK	no/yes	no	average of the sum of the local gradient	intensity comparison of pair-wise pixels in retina sampling pattern
BinBoost	no/yes	no	no	projection by learned hash function
LATCH	no/yes	no	no	intensity comparison of patch triplet arrangements

potentially sensitive to edge responses. To be invariant to rotation, an orientation is assigned to the obtained stable points according to the local gradient orien-

tation histogram within a region around the point. In addition, it accumulates the orientations of a 16×16 neighborhood of sample points around the salient point location into orientation histograms by summarizing the contents over 4×4 sub-regions. A 128-dimensional descriptor vector is finally generated to represent each point.

2.3.2 SURF (detector/descriptor)

SURF is an efficient and robust scale and rotation-invariant method proposed by Bay et al. [12] with the aim for fast salient point location and descriptor generation. SURF is based on a Hessian matrix, where the components of the Hessian matrix are generated by convolution of the Gaussian second-order derivative with the image pixels. Box-filters together with integral images are exploited to approximate the Hessian matrix which is used to measure the salient points. The Gaussian scale space of SURF is established computationally efficiently by up-scaling the size of the box-filter. The extrema of the determinant of the Hessian matrix are selected as salient points and the scale and location are updated through an interpolating process. Each of the obtained salient points is assigned an orientation which is estimated by summing the horizontal and vertical Haar-wavelet responses within a sliding orientation window covering an angle of 60 degrees. For the SURF descriptor generation, first the square region centered on and oriented along the salient point is divided into a number of 4×4 sub-square regions. Then, it calculates the value and absolute value of Haar-wavelet responses along horizontal and vertical directions within each sub-region. Finally the total 64-dimensional $(4 \times 4 \times 4)$ descriptor can be generated efficiently by making use of the integral image.

2.3.3 MSER (detector)

Maximally stable extremal regions (MSER), proposed by Matas et al. [98], is an affine invariant region detector. MSER computes the connected binary regions through a large set of multiple thresholds, and the selected regions are those

that maintain unchanged shapes over a range of thresholds. During the affine invariant regions detection, the area of each connected component is stored as a function of intensity and the "maximally stable" ones are selected as candidates by analyzing the changes of function values for each potential region. The final maximally stable extremal regions are the ones that maintain an unchanged or similar function value over a large range of multiple thresholds. The shape of each obtained region is further estimated by elliptical regions by computing the eigenvectors of their second-moment matrix. Then the local neighborhoods are normalized into circular regions to achieve affine invariance.

2.3.4 HESSIAN-AFFINE (detector)

The Hessian-Affine region detector proposed by Matas et al. [97] is based on the Hessian matrix. A related variant of the Hessian-Affine detector is the Harris-Affine detector which employs the Harris detector to find the salient points. Since the second derivatives in the Hessian matrix offer strong responses on blobs and ridges, the extrema of the determinant of the Hessian matrix are searched by applying non-maximum suppression using a 3×3 window over the entire image. To deal with the scale invariance, given an extremum location, a scale-dependent signature function is defined on its local neighborhood and the corresponding scale can be determined by searching for scale-space extrema of the signature function. The estimation of the affine shape is applied to each extremum and an elliptical region is fit around each point using the second moment matrix of the intensity gradient. Finally, the affine region is normalized into a circular region. In this chapter, the improved Hessian-Affine detector [114] is used, which proposes the gravity vector assumption to fix rotation uncertainty.

2.3.5 FAST (detector)

The high-speed corner point detector named features from accelerated segment test (FAST) was proposed by Rosten and Drummond [96]. The simple scheme of FAST corner detection is based on a circle (the radius of the circle is three

pixels) of sixteen pixels around the candidate point. If there exists a set of twelve contiguous pixels in the circle which are all brighter or all darker than the intensity of the candidate point pixel value plus a threshold, the point will be classified as a corner point. However, this scheme has a limitation for sampling less than twelve pixels and the efficiency of the corner detector depends on the distribution of corner appearances. To overcome the above weaknesses, a machine learning approach is employed on training sets to establish a decision tree for fast and accurate corner detection. Moreover, the issue of multiple features being detected adjacent to one another, can be solved by applying non-maximum suppression on the detected candidate corner points.

2.3.6 CenSurE (detector)

The scale invariant center-surround salient point detector (CenSurE) is proposed by Agrawal et al. [115]. CenSurE determines the salient points by exploiting the extrema of the Hessian-Laplacian matrix across all scales and locations. Inspired by SIFT which uses the Difference of Gaussian function to approximate the Laplacian of Gaussian function, CenSurE employs a simplified center-surround filter called bi-level filter to approximate the Laplacian of Gaussian for fast computation. The CenSurE detector computes the response of the bi-level filter at all locations and all scales, and detects the extrema in a local neighborhood (based on the non-maximum suppression method, which is the same as SIFT and SURF). For each obtained extremum, the accurate location of the potential points can be determined directly, since the responses are calculated on the original image. Furthermore, through computing the Harris measure for the potential points, those points with weak corner responses will be eliminated.

2.3.7 GFTT (detector)

Good feature to track (GFTT) is a salient point detector proposed by Shi and Tomasi [116], which is derived from an image motion model. GFTT is used as a method for feature selection, tracking and monitoring, and it performs well

under affine image transformations. According to the proposed feature selection criteria, a candidate point is accepted if it is defined as a good feature which can be tracked well. GFTT is based on the Harris corner detector and it defines points with large eigenvalues of a special matrix as corners. To ensure the robustness of corners, potential corners with minimum eigenvalues less than a threshold are eliminated. Candidates which are closer than a certain distance-threshold to a strong corner are also rejected.

2.3.8 KAZE (detector)

Most salient point approaches (SIFT, SURF) construct the scale space based on linear multi-scale Gaussian pyramids. However, the Gaussian function does not respect the natural boundaries of objects and smoothes the details and noise at the same level, which leads to loss of localization accuracy and distinctiveness. The use of a nonlinear scale space is expected to reduce noise but to retain the object boundary structure in order to obtain accurate positions of salient points. The traditional method is based on the forward Euler scheme for solving nonlinear diffusion but requiring significant computational complexity. Therefore, the nonlinear scale space in KAZE [106] proposes to use the additive operator splitting algorithm (AOS) for efficient nonlinear diffusion filtering. The framework of KAZE first convolves the image with a Gaussian kernel of standard deviation, and then builds the nonlinear scale space in an iterative way using the AOS scheme. Based on the response of the scale-normalized determinant of the Hessian matrix at multiple scale levels, the extrema responses can be detected as salient points by non-maximum suppression and the position of the salient points can be estimated with sub-pixel accuracy using quadratic fitting.

2.3.9 BRIEF (descriptor)

Binary robust independent elementary features (BRIEF), designed by Calonder et al. [117], uses an efficient binary string descriptor to represent the salient points. With regard to the BRIEF descriptor generation, Gaussian smoothing

is first utilized to reduce the effect of noise sensitivity such that it can achieve good performance in complex scenes. The value of each bit in the binary string depends on the intensity comparison of two points inside the local patch centered on each salient point (provided by detectors, as BRIEF is a descriptor), i.e., if the value of first point is larger than the second then it is set to "1", otherwise to "0". The pixel-pairs sampling patterns are randomly selected using a Gaussian distribution (locations that are closer to the center of the patch are preferred) around the smoothed patch center. Similarity of two binary string descriptors is calculated using the Hamming distance, which is significantly more efficient than the common Euclidean distance. The BRIEF descriptor is not rotation invariant.

2.3.10 ORB (detector/descriptor)

ORB (oriented FAST rotated BRIEF) [118] is a combination of the FAST detector and the BRIEF descriptor. The ORB detector applies the FAST corner detector to find potential salient points. However, FAST does not offer scale information, and has large responses along edges. ORB builds a scale pyramid of the image and keeps the top N number of keypoints by the Harris corner measure at each level in the scale pyramid. The scale information is the scale factor of the specific level of the image pyramid. The direction of points is computed using their intensity centroid [15]. The intensity centroid approach assumes that the intensity of a keypoint is offset from its center, and it can be used to compute the moments of a patch and also to find its centroid. The orientation is defined as the direction of the vector between the keypoint location and the centroid position in the patch. The generation of the ORB binary string descriptor also uses the comparison of intensities of pixel-pairs based on the oriented BRIEF descriptor. Additionally, a combination of earning and greedy search is introduced for de-correlating BRIEF features under rotational invariance, leading to a better performance.

2.3.11 BRISK (detector/descriptor)

In the implementation of BRISK [16], the scale space is also based on the simple image pyramid. For the salient points detection, BRISK first employs AGAST [119] which is essentially an extension for accelerated performance of the FAST detector to locate the potential keypoints at each layer in the scale space. Then it measures their saliency via comparing FAST scores with respect to its eight neighbors in the same layer and 3×3 neighbors in the layer above and below. The local maxima of FAST score points will be identified as salient points. The accurate location and scale of each salient point are obtained in the continuous domain via refinement of quadratic function fitting. BRISK presents a novel sampling pattern which consists of sample points equally distributed on concentric circles centered around the salient point. It weights each respective circle in the pattern with a standard deviation Gaussian, and then divides all the sampling-point pairs in the pattern into short-distance pairs and long-distance pairs based on the defined threshold. The direction of the patch is determined via the average of the sum of the local gradients of all selected long distance pairs. The bit-vector descriptor is assembled by comparing all the short-distance pair-wise intensities.

2.3.12 FREAK (descriptor)

Similar to the BRISK scheme which uses a pre-defined pattern to estimate the orientation and for generating the binary string features, the FREAK [17] descriptor is based on the retina sampling pattern. The retina sampling pattern simulates the distribution of ganglion cells over the retina which reduces exponentially with the distance to the center. The orientation is calculated mainly based on selected pairs with symmetric receptive structure with respect to the center point of the patch. The direction of the patch is also obtained by averaging the sum of the local gradient of the defined pairs in the structure. In the descriptor creation of FREAK, less correlated pairs over a retina pattern are selected based on a similar learning algorithm performed in ORB and the intensities are then compared to generate the binary strings.

2.3.13 BinBoost (descriptor)

The approach of BinBoost is a supervised learning framework to generate a low dimensional but highly discriminative local binary representation. A hash function is implemented as a sign operation on a linear combination of non-linear weak classifiers which are gradient based image features, and the hash function is learned by the optimization of a loss function with the aim to reduce the Hamming distances between binary representations of similar patches in training data, while increasing the Hamming distances between binary representations of dissimilar patches in the training data.

2.3.14 LATCH (descriptor)

LATCH extracts learned patch triplet arrangements in a salient region, and compares the intensity of the triplet patches to form the binary string codes. The learning procedure of LATCH is based on training data with labels, and possible triplet arrangements are extracted from the training data. It defines the quality of an arrangement by summing the number of times it correctly yielded the same binary value for positive pairs and different values for negative pairs. A candidate arrangement is selected, if its absolute correlation with all previously selected arrangements is smaller than a certain threshold such that the obtained triplet arrangements are with less correlation.

2.4 Fully Affine Space Framework

The main idea behind the framework of fully affine space is that the projective transformation induced by camera motion around a smooth surface can be approximated by an affine transformation, and it consists of all possible affine distortions caused by the change of the camera's optical axis orientation from a frontal view. The reason to employ this scheme is that we expect two salient points to be correctly matched under certain perspective transformations. The

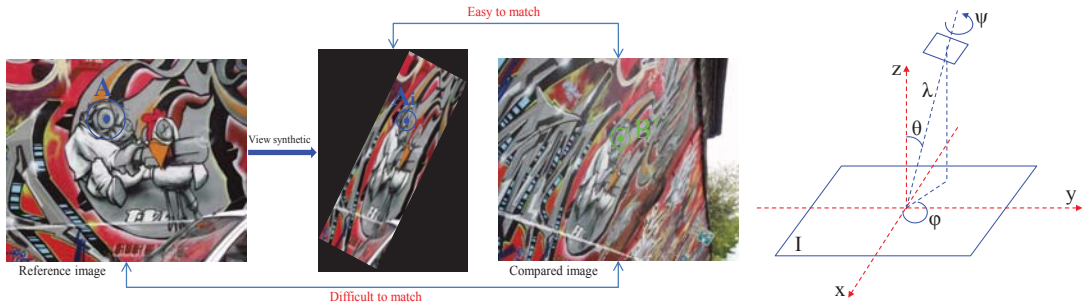


Figure 2.1: (a) Illustration of the synthetic view generation for correct correspondence matching. (b) Illustration of the camera model under affine transformation.

fully affine space framework could also be viewed as a data augmentation technology which expands the training data by systematically adding transformed samples. The transformed samples are typically generated to be label-preserving such that they can encourage the system to become invariant to different transformations. As illustrated in Figure 2.1 (a), it is difficult to match point A in the reference image to point B' in the compared image, but it is easy to match point A_i which is located in the deformed view image arising from viewpoint changes to point B'. Generating a deformed view image can be modeled by an affine transformation of the original image, where the affine transformation can be decomposed into a zoom, rotation, tilt, and rotation around the optical axis [120].

$$\begin{aligned} A &= \lambda R(\psi) T_t R(\varphi) \\ &= \lambda \begin{bmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\varphi) & -sin(\varphi) \\ sin(\varphi) & cos(\varphi) \end{bmatrix} \end{aligned} \tag{2.1}$$

where $\lambda > 0$ is a zoom factor, $R(\psi)$, $R(\psi)$ are rotations and t is the tilt, as shown in Figure 2.1 (b). The parameter $\psi \in [0, 2\pi)$ denotes the angle of planar rotation around the optical axis. The angle θ between the z axis and the optical axis is called the latitude and $t = 1/cos(\theta)$. The angle $\varphi \in [0, \pi)$ between the x axis and the projection of the optical axis is called the longitude. Then, each synthesized view can be described by the parameters of λ (zoom), $R(\psi)$

(planar rotation), $t = 1/cos(\theta)$ (the rotation angle of the latitude) and $R(\varphi)$ (the rotation angle of the longitude). The simulated latitudes θ correspond to tilts $t = 1, a, a^2, ..., a^n$, with $a > 1$, and a is set $\sqrt{2}$ for a good compromise between accuracy and efficiency. Each tilt in the fully affine space is a t sub-sampling. The number of rotated images for each tilt is *2.5t*. Thus, the complexity is proportional to the amount of tilts. As the fully affine space can significantly increase the precision of correspondence matching, we integrated the recent salient point methods with the fully affine space framework and evaluated their accuracy and efficiency.

Generally, the Nearest Neighbor Distance Ratio (*NNDR*) is used as the matching strategy to find the similar descriptors in the image pairs. *NNDR* defines that two points will be considered to be matched only if $||D_A - D_B||/||D_A - D_C|| < threshold$, where D_B is the first and D_C is the second nearest neighbor to D_A. However, for the matched correspondences in the specific fully affine space, lots of repeatable salient points are present in the synthetic view images which results in the *NNDR* to be close to one for some correct correspondences, thus, those correct correspondences will be easily defined as false according to the threshold (less than one) of *NNDR*. In order to address this issue, we propose to use the *K-order NNDR* matching strategy for correspondence matching in the fully affine space. Unlike the standard *NNDR* which only takes the first and second nearest neighbors into account, *K-order NNDR* fully explores the relationship among the group of K nearest neighbors, such that it can address the problem faced by *NNDR* but without increasing the computational cost. The *K-order NNDR* is characterized as follows:

$$K\text{-}order\ NNDR = R_k \times (1 - \frac{w}{\prod_{i=2}^{k-1} R_{i-1}}) \tag{2.2}$$

where $R_k = ||D_A - D_1||/||D_A - D_k||$ and D_k is the k^{th} nearest descriptor to D_A. w is a weight which is set to 0.01 in the experiments to achieve good performance.

2.5 Experimental Setup

The experimental environment for the evaluation is a Intel Quad Core i7 Processor (2.67GHz), 12GB of RAM, 64 bit OS. The implementations of Hessian-affine, KAZE, LATCH and BinBoost are from the authors, others are implementations from OpenCV. The parameters of each salient point method were set to the defaults and we used 8 randomized forests in the KD-tree index, 20 hash tables in the multi-probe LSH index. Our evaluation implementations are available at: http://press.liacs.nl/researchdownloads/.

2.5.1 Datasets

The performance of salient point detectors and descriptors is evaluated on the Oxford dataset proposed by Mikolajczyk and Schmid [108] and the dataset designed by Fischer et al. [121]. The Oxford dataset contains eight groups, and each group consists of six image samples (a total of 48 images) with various transformations (rotation, viewpoint, scale, JPEG compression, illumination and image blur). The Fischer dataset is a large scale dataset that includes 16 groups and each group contains 26 images generated synthetically by applying 6 types of transformations (zooming, blurring, illumination, rotation, perspective and nonlinear). Some examples of each dataset used for evaluation are illustrated in Figure 2.2.

2.5.2 Evaluation Criteria

The criteria employed to measure the performance of the salient point methods in each application are summarized in Table 2.2. We follow the commonly used evaluation protocol [87, 107, 108, 122]. The score of repeatability, recall and precision, and the number of correct correspondences are used as evaluation criteria in the experiments.

(a)

(b)

Figure 2.2: Examples from each dataset for the evaluation of salient point methods. (a) Examples from the Oxford dataset [108] used for the evaluation of the accuracy of correspondence matching. (b) Examples from the Fischer dataset [121] used for the evaluation of the accuracy of correspondence matching.

Table 2.2: Overview of the evaluation criteria used in the experiments.

Criteria	Function description
Repeatability [107]	Measures the performance of the detector: the higher the repeatability score, the better the performance.
Recall and precision [108]	Measures the accuracy of correspondence matches: a distinctive descriptor shows high recall at any precision.
Number of correct correspondences	Total amount of correct correspondences between two compared images, a robust method shows a high score.

2.6 Results and Discussions

2.6.1 Detector Evaluation

In this section, we test the performance of each salient point detector on the benchmark Oxford dataset [108] and the Fischer dataset [121]. The evaluated salient point detectors are: SIFT, SURF, ORB, BRISK, FAST, CenSurE, GFTT,

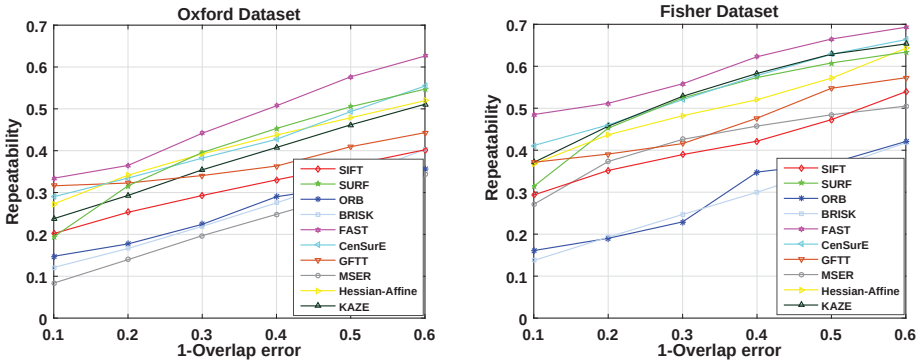


Figure 2.3: The performance evaluation of salient point detectors based on the criterion of repeatability.

KAZE, MSER and Hessian-Affine.

An important evaluation criterion from the research literature is repeatability [107]. The repeatability score is calculated as the ratio between the number of correspondences and the minimum of m_1 and m_2, where m_1, m_2 denote the number of points in the reference and the query images after projecting the reference image points by the ground truth homography and discarding those points outside the common area, respectively.

$$repeatability = \frac{C(m_1, m_2)}{min(m_1, m_2)} \tag{2.3}$$

$C(m_1, m_2)$ is the number of correspondences between m_1 and m_2. An overlap error is used to identify the correspondence. For a keypoint region in the query image which is the nearest one to a projection keypoint region in the reference image by using homography: if the ratio between the intersection of the two regions and the union of the two regions is larger than the overlap error, it will be considered as a correspondence. We compute the average repeatability scores on the whole dataset, respectively, thus, the detection performance of each method can be estimated in a comprehensive perspective. The trend of average repeatability under varying overlap errors (in the range from 0.4 to 0.9) is shown in Figure 2.3.

The evaluation results based on the two datasets illustrate that an increase in

the repeatability scores is clearly indicated when the value of 1-overlap error becomes larger. We can also notice that the FAST detector had the highest repeatability and the ORB and BRISK detectors obtained the lowest scores. The detectors SURF, Hessian-Affine, KAZE, and CenSurE have a similar rank on both datasets. The performance of the nonlinear scale space detector KAZE reveals superior results to the well known SIFT detector. All detectors can reach a stable and acceptable performance when the value of overlap error is 0.5, so the overlap error will be set at 0.5 to identify the correspondences in the following experiments.

Since the salient point detection mechanism in each salient point method is based on a different scheme, which results in a different computational complexity, and a different set of feature points can be extracted from the same image, time costs should be compared statistically. We applied different types of detectors to various test images, in order to determine statistically significant results. The average number of detected points and the time cost of the compared salient point methods are shown in Table 2.3.

Table 2.3: Comparison of average number of detected points and detection time

Method	Oxford Dataset [108]		Fischer Dataset [121]	
	Average number of points	Time cost(ms) of 1000 points	Average number of points	Time cost(ms) of 1000 points
SIFT	5472	40	5607	52.02
SURF	5368	22.8	6138	34.8
ORB	497	27.0	490	29.5
BRISK	1498	20.2	1607	19.3
FAST	15857	0.31	17388	0.27
CenSurE	915	20.1	920	25.1
GFTT	1000	31.2	984	35.6
MSER	750	341.8	793	360.5
HESSIAN-AFFINE	3680	247.8	3693	260.1
KAZE	2940	59.8	3108	73.5

The results listed in Table 2.3 reveal that the most efficient detector is FAST. FAST detected the largest number of salient points on both datasets, which is almost ten times higher than what was obtained by other detectors. FAST defines the salient points according to simple intensity comparisons, thus, the time cost is only 0.31 ms for a total of 15857 points on the Oxford dataset [108] and 0.27 ms for 17388 points on the Fischer dataset [121]. The most time-consuming detectors are MSER and Hessian-Affine, because they need do the ellipse fitting for each salient point. The detectors SIFT, SURF, ORB, BRISK and KAZE all contain scale space and rotation estimation procedures. KAZE builds the nonlinear scale space in an iterative way using the AOS scheme which is much more time consuming than the linear scale space calculation. As SURF, ORB and BRISK speed up building the scale space, they are more efficient than the SIFT detector.

2.6.2 Descriptor Evaluation

The Oxford and Fischer datasets are also utilized in the local descriptors evaluation. Note that some of the salient point detectors from the previous section do not define descriptors and are not compared here. In order to make an objective comparison of different salient point descriptors, SURF was applied as the salient point detector, as the SURF detector is scale invariant and it provides a high repeatability score according to its performance in the detector evaluation. We combined SURF detectors with local descriptors including SIFT, SURF, ORB, BRIEF, BRISK, FREAK, BinBoost and LATCH. The evaluation starts by extracting salient point features from the reference images and establishing a KD-tree or LSH index space for the obtained local features. Then, we extract features from the query image and match them against the features from each reference image based on the approximate nearest neighbor search. In the matching procedure, a KD-tree index is established for real value descriptors and the Euclidean distance is used for matching, while binary string descriptors are matched in an LSH index using the Hamming distance.

The *NNDR* is used as the matching strategy to find similar descriptors in image pairs. In addition we use recall and 1-precision [108] (not to be confused with precision@1) as criteria to measure the performance of various salient point descriptors. Recall denotes the number of correct matches with respect to the number of correspondences between two compared images, and the precision is the number of correct matches with respect to the total number of matches.

$$recall = \frac{\#correct\_matches}{\#correspondences} \tag{2.4}$$

$$precision = \frac{\#correct\_matches}{\#total\_matches} \tag{2.5}$$

We varied the value of the threshold in the *NNDR* to obtain the curves of the tendency of the average recall vs. 1-precision under each transformation. Figure 2.4 and Figure 2.5 show the results on each dataset. We also provide the area under the recall vs. 1-precision curve, averaged over all image transformations in each dataset, as shown in Table 2.4 and Table 2.5. A distinctive descriptor would give a high score of area under each curve (AUC).

Table 2.4 and Table 2.5 summarized the results of AUC under each transformation as well as the average score. SIFT, BRISK, and FREAK show good performance for all image degradations on the two datasets. Looking at the performance on the Oxford dataset [108], all descriptors perform better on image changes (blur, illumination and JPEG compression) than on affine deformation changes (rotation, scale and perspective). The descriptors created by SIFT, BRISK, FREAK, SURF, and BinBoost are more robust and distinctive than ORB, BRIEF and LATCH under affine deformation. This is mainly because the BRIEF descriptor only conducts pixel-pair intensity comparisons and is not rotation invariant, while the ORB descriptor as an improved BRIEF descriptor is rotation invariant and resistant to noise, but not scale invariant. The LATCH descriptor uses the same scale information causing it not to be scale invariant. For the scores under changes of blur and JPEG compression, the BinBoost descriptor obtains the lowest score, thus, it is more sensitive to those types of noise. An illumination change

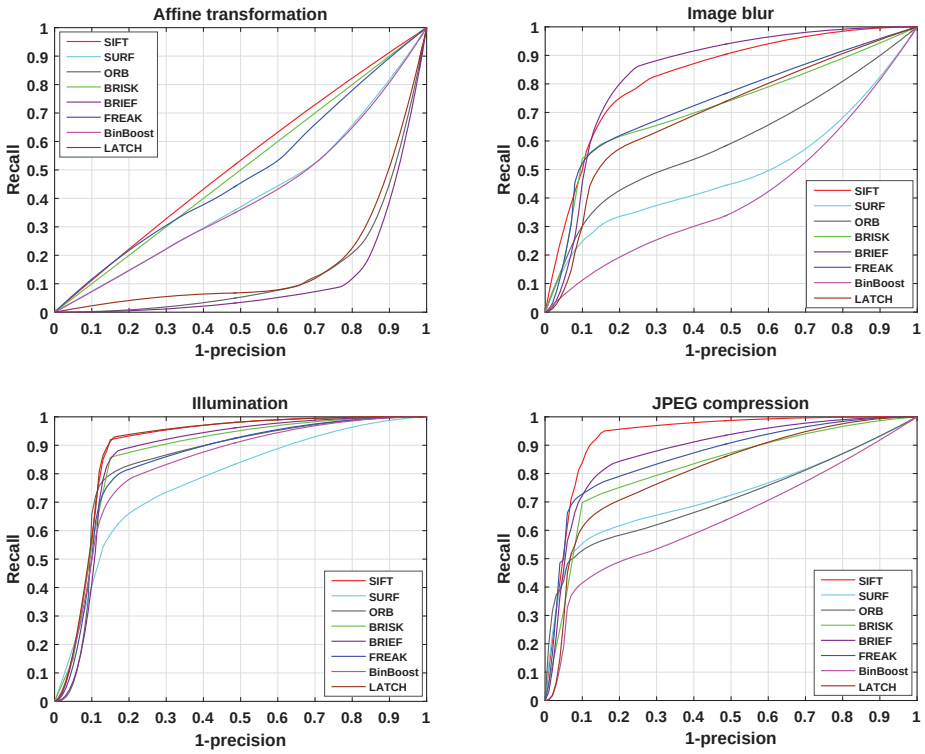


Figure 2.4: Comparison of various descriptors using recall vs 1-precision under different image degradations. The evaluation results are for the Oxford dataset [108].

has a big influence on the SURF descriptor, while the other descriptors are robust to illumination changes and show scores close to each other. The evaluation results on the Fischer dataset [121] show the same tendency under the changes of image blur and perspective when compared to the results on the Oxford dataset [108]. In addition, the descriptors of ORB, BRIEF and LATCH also show their weakness under the change of image zoom.

The time and memory complexity of local descriptor extraction is also statistically analyzed in this section. The average time costs for generating local descriptors based on the Oxford dataset [108] and the Fischer dataset [121] are shown in Table 2.6. It is clear that binary string descriptors are more efficient than real valued descriptors in terms of memory requirement. The SIFT descriptor has the highest time complexity, followed by the BinBoost descriptor. The SURF

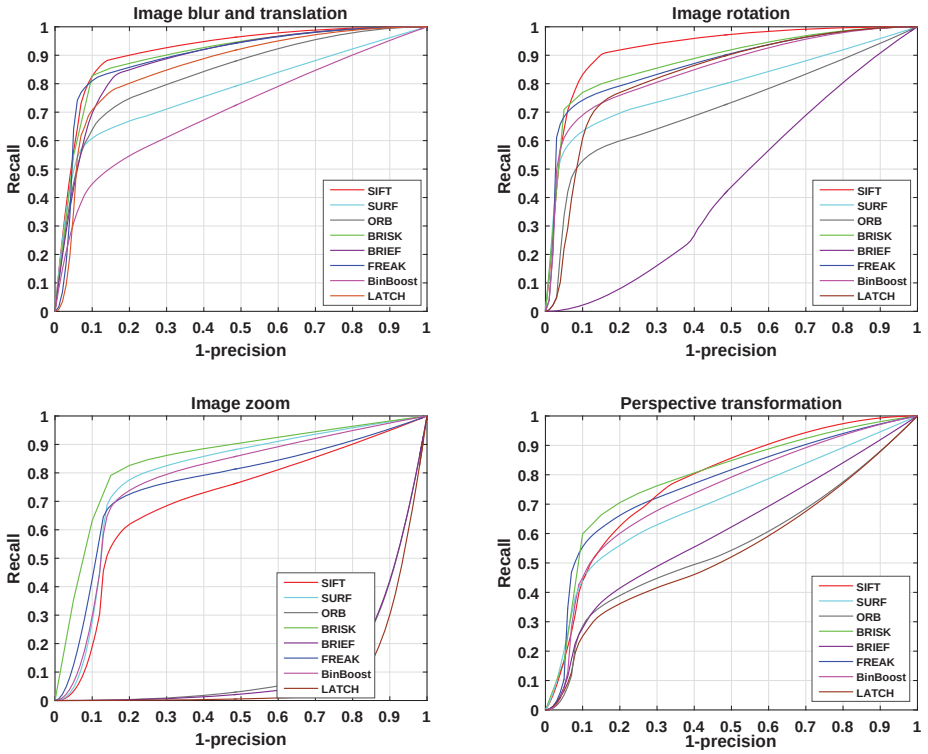


Figure 2.5: Comparison of various descriptors using recall vs 1-precision under different image degradations. The evaluation results are for the Fischer dataset [121].

descriptor is more efficient than the SIFT descriptor. However, binary string descriptors like ORB, BRIEF, BRISK and FREAK perform much faster than the other local descriptors. Thus, the binary string descriptors ORB, BRIEF, BRISK and FREAK are more appropriate for real-time applications.

2.6.3 Affine Invariant Evaluation

According to the above performance evaluation, most of the salient point methods are significantly influenced by affine transformations. As the framework of fully affine space could improve the accuracy of correspondence matching under huge viewpoint changes,we evaluate each salient point method in the framework of fully affine space and employ the proposed *K-order NNDR* matching strat-

egy to define the final correspondences. The evaluated salient point methods in the framework of fully affine space contain SIFT+SIFT (detector+descriptor), SURF+SURF, SURF+BRIEF, ORB+ORB, BRISK+BRISK, SURF+FREAK, SURF+BinBoost and SURF+LATCH. We also use randomized KD-trees to establish an index space and Euclidean distance for real valued descriptor matching. Binary descriptors are matched in a LSH index space with Hamming distance.

For the extracted local features of salient points in two compared images I and I', the obtained set of matches can be defined as:

$$M_{I-I'} = \{p_I^i \leftrightarrow p_{I'}^j\} \tag{2.6}$$

point $p_{I'}^j$ in image $I^{'}$ is the closest neighbor to point p_I^i in image I. We need to note the situation that the same point in the index space could be the nearest neighbor to different points in the query space (many-to-one matches), we then enforce a one-to-one constraint through a cross-check operation. The cross-check operation starts by building an index space for the local descriptors in the query image, and searching the k closest neighbors for each point in the reference image. Then we build the index space for the local descriptors in the reference image, and find k nearest neighbors for each point in the query image. Only if they

Table 2.4: The Oxford benchmark results [108]. Numerical results summarizing area under the recall vs. 1-precision curve for different transformations. Higher results are better.

Descriptor	Affine	Blur	Illumination	JPEG	Average
SIFT	**0.523**	0.832	0.892	**0.931**	**0.794**
SURF	0.404	0.49	0.774	0.723	0.598
ORB	0.141	0.596	0.844	0.711	0.573
BRISK	0.5	0.716	0.866	0.824	0.727
BRIEF	0.113	**0.841**	0.864	0.879	0.674
FREAK	0.484	0.735	0.843	0.863	0.731
BinBoost	0.4	0.412	0.83	0.641	0.571
LATCH	0.164	0.697	**0.894**	0.809	0.641

Table 2.5: The Fischer benchmark results [121]. Numerical results summarizing area under the recall vs. 1-precision curve for different transformations. Higher results are better.

Descriptor	Blur+Translation	Perspective	Rotation	Zoom	Average
SIFT	**0.915**	0.776	**0.925**	0.705	0.83
SURF	0.777	0.702	0.791	0.796	0.766
ORB	0.837	0.556	0.715	0.128	0.559
BRISK	0.902	**0.79**	0.887	**0.85**	**0.857**
BRIEF	0.882	0.606	0.443	0.117	0.41
FREAK	0.893	0.767	0.871	0.763	0.824
BinBoost	0.707	0.735	0.85	0.78	0.768
LATCH	0.859	0.54	0.832	0.1	0.583

satisfy formula (2.7), they can be considered a match.

$$M = \{M_{I-I'} = \{p_I^i \leftrightarrow p_{I'}^j\} \ \wedge M_{I'-I} = \{p_{I'}^j \leftrightarrow p_I^i\}\} \tag{2.7}$$

We use the proposed *K-order NNDR* matching strategy, replacing the original *NNDR* matching strategy, to define the matched correspondences:

$$C = \{p_I^i \leftrightarrow p_{I'}^j | \textit{K-order NNDR}(p_I^i, p_{I'}^j) < \textit{threshold}\} \tag{2.8}$$

where *K-order NNDR*$(p_I^i, p_{I'}^j)$ denotes that two similar descriptors satisfy the *K-order NNDR* threshold and $(p_I^i, p_{I'}^j) \in M$.

As the salient point extraction in the fully affine space could result in duplicate correspondences, we eliminate these duplicates according to the spatial distance (2

Table 2.6: Comparison of average description time cost on both two datasets

Method	Feature dimensions	Memory requirement (1000 points)	Oxford Dataset [108] Average time cost(s)/5400	Fischer Dataset [121] Average time cost(s)/6000
SURF+SIFT	128 float	0.488M	4.3	4.8
SURF+SURF	64 float	0.244M	0.24	0.26
SURF+BRIEF	256 bit	0.03M	0.013	0.015
SURF+ORB	256 bit	0.03M	0.015	0.018
SURF+BRISK	512 bit	0.06M	0.028	0.032
SURF+FREAK	512 bit	0.06M	0.02	0.025
SURF+BinBoost	256 bit	0.03M	3.03	3.27
SURF+LATCH	256 bit	0.03M	0.25	0.28

pixels) of point location in both image. To further determine whether a matched correspondence is correct or not, each correspondence obtained by the *K-order NNDR* is determined as correct only if its corresponding point is geometrically the closest point within the defined pixel coordinate error, and the final correct correspondences are evaluated by the ground-truth homography:

$$Correct\_matches = \{p_I^i \leftrightarrow p_{I'}^j | D(H(p_I^i), p_{I'}^j) < \varepsilon\} \tag{2.9}$$

where $D(H(p_I^i), p_{I'}^j)$ is the position error after the ground-truth homography H projection for the point in image I, and in all cases, the ε is set as 2 pixels.

Following common practice in evaluation protocols, we use the total number of correct matches between two compared images as criterion for the evaluation of correspondences matching. As ASIFT set the *NNDR* matching threshold to 0.73×0.73, we use the same threshold in our *K-order NNDR*. Moreover, in the framework of fully affine space, the parameter of tilt t controls the number of generated synthetic images in the affine space, and we need to note that larger value of the parameter t leads to higher computational complexity of the framework of fully affine space. For the evaluation, we set the parameter of t to 5, 6, and 7 corresponding to the numbers of the generated synthetic images 27, 41, and 61, respectively.

2.6.3.1 Parameter of K in *K-order NNDR*

In this part, we evaluate the impact of size K in the *K-order NNDR*. The images under viewpoint changes in the Oxford dataset [108] and the images for perspective changes in the Fischer dataset [121] are used. The impact of K in the *K-order NNDR* is shown in Figure 9. The test is based on the SIFT+SIFT, where the tilt in the scale space is set to 5. Figure 9 displays that the amount of correct correspondences shows a tendency to increase when K becomes larger, and for the SIFT detector with the SIFT descriptor, the *K-order NNDR* shows superior results to the original *NNDR*. Since the increase of magnitude of the correct correspondence is not significant when K varies from 4 to 6 and larger

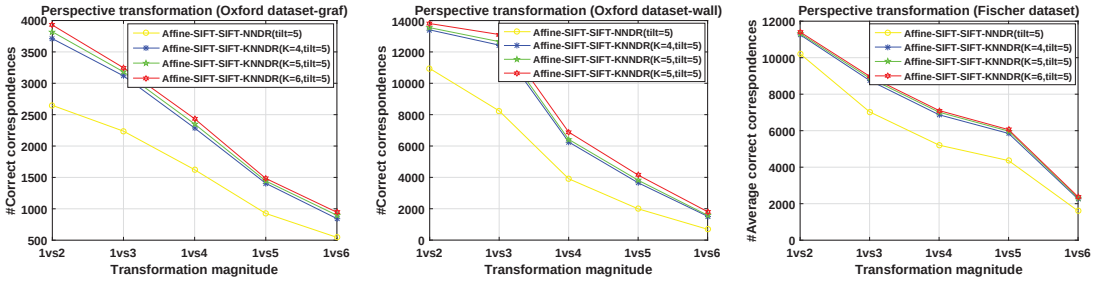


Figure 2.6: The demonstration of parameter K in the *K-order NNDR* (*KNNDR*) used in the fully affine space framework.

value of K reduces the efficiency of *K-order NNDR*, we set K equal to 4 in the following experiments.

2.6.3.2 Correspondence Matching Using the Framework of Fully Affine Space

For an objective comparison, we first evaluated the performance of each method without using the fully affine space framework. Figure 2.7 displays the amount of correct correspondences on the Oxford dataset, as well as the average numbers of correct correspondences on the Fischer dataset. It is clear that the SIFT+SIFT performs best on both datasets, and ORB+ORB, BRISK+BRISK are more sensitive to the affine changes (scale, rotation and perspective changes) than the other salient point methods. However, when the magnitude of perspective transformation becomes larger, all methods show poor performance.

As all salient point methods can only tolerate a small magnitude of viewpoint transformation, we apply the fully affine space framework and the proposed *K-order NNDR* scheme to evaluate their performance. Figure 2.8, Figure 2.9 and Figure 2.10 depict the evaluation results for real valued and binary string descriptors. It can be observed that a similar tendency is demonstrated on both datasets. When comparing the results of salient point methods using the fully affine space framework with the previous results, the performance has been significantly improved under large viewpoint transformations. We can note that Affine-

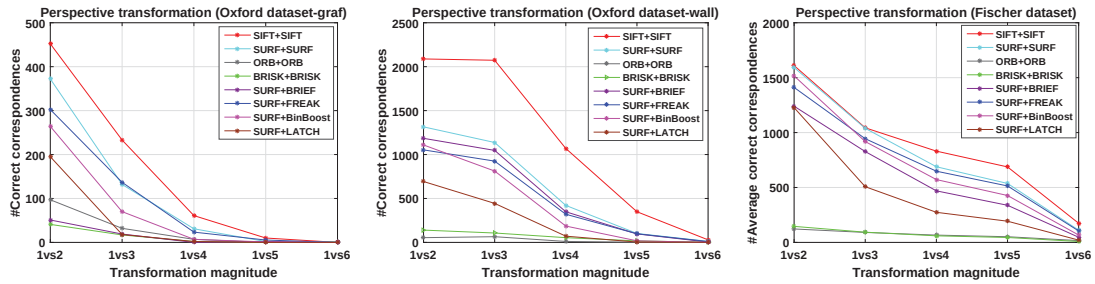


Figure 2.7: The demonstration of the amount of correct correspondences under perspective changes for each salient point method without using the fully affine space framework.

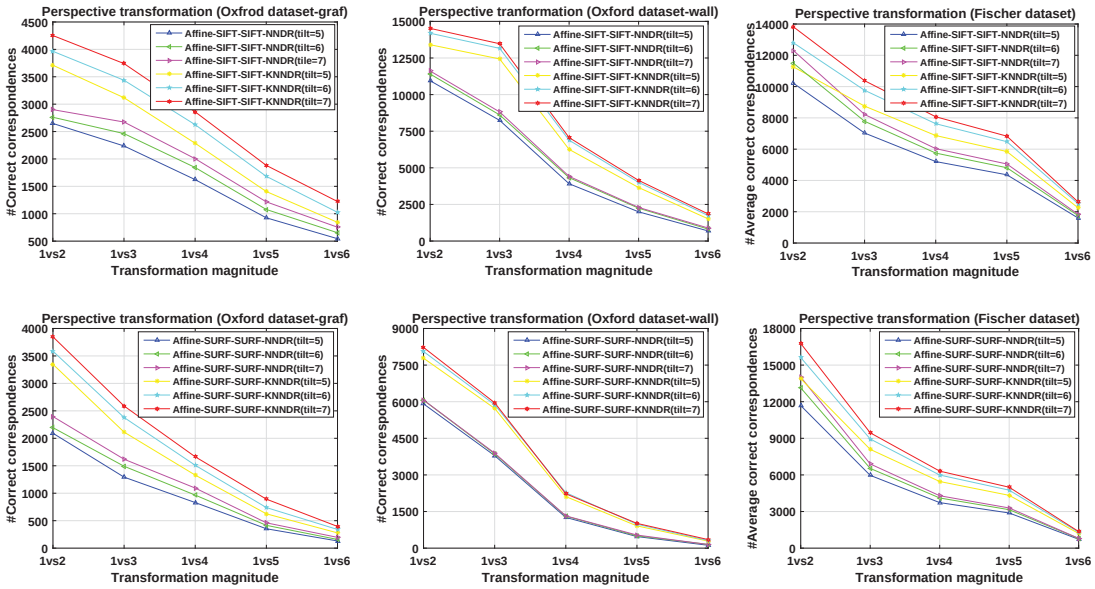


Figure 2.8: Evaluation results of salient point methods with real valued descriptor. The fully affine space framework is applied (the tilt varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

SIFT+SIFT obtained the highest number of correct matches in all cases, and this is mainly due to the distinctiveness of the SIFT local descriptor. Moreover, the real valued descriptors are more distinctive than binary string descriptors.

In addition, for the comparison between *NNDR* and *K-order NNDR*, the evaluation results show the advantages of the *K-order NNDR* matching strategy. The

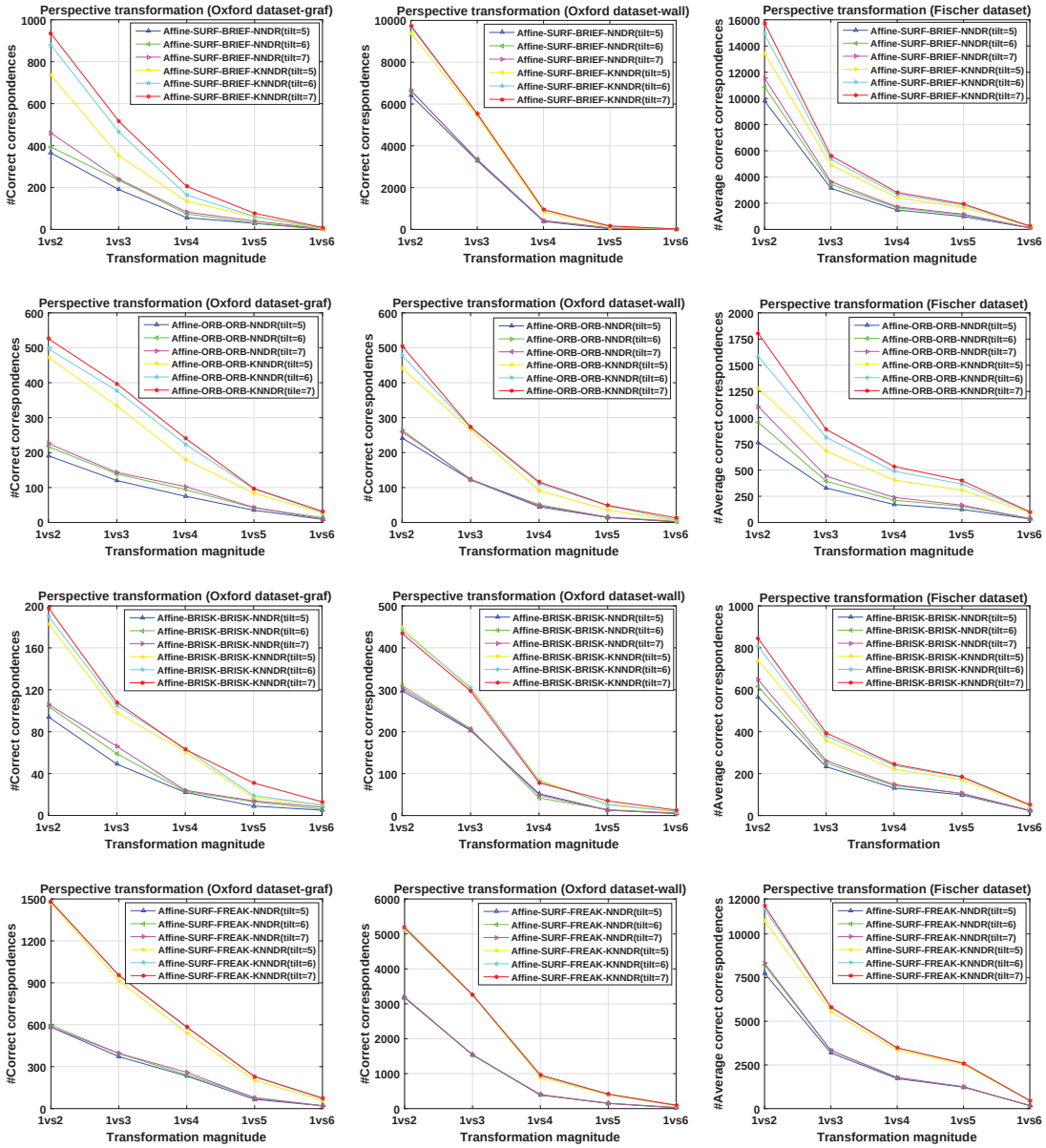


Figure 2.9: Evaluation results of salient point methods with hand-crafted binary string descriptor. The fully affine space framework is applied (the tilt varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

K-order NNDR is effective for all the salient point methods. We can observe that *K-order NNDR* finds roughly double the number of correct correspondences compared to the original *NNDR*. Moreover, the results of *K-order NNDR* with tilt

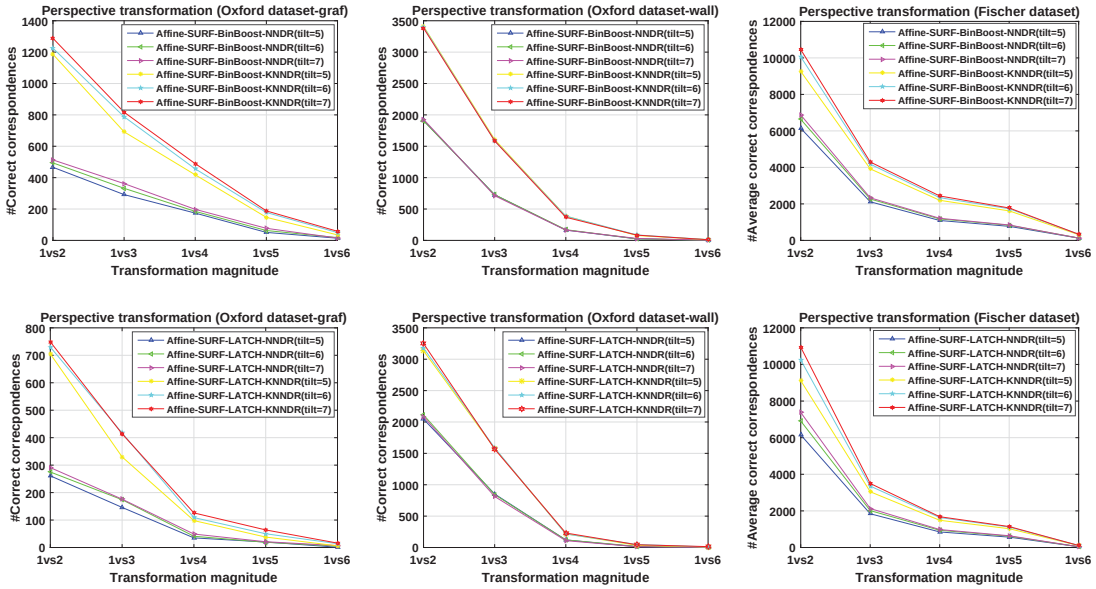


Figure 2.10: Evaluation results of salient point methods with supervised learning based binary string descriptors. The fully affine space framework is applied (the tilt is varied from 5 to 7), and both *NNDR* and *K-order NNDR* (*KNNDR*) are compared.

equal to 5 is even much better than *NNDR* with tilt equal to 7. This means that *K-order NNDR* can get high accuracy even at a low computational complexity of the fully affine space framework. Although the discrimination of binary string features is insufficient, binary string descriptors using the *K-order NNDR* can also offer competitive results compared to real valued descriptors using *NNDR*.

According to the above evaluation results on both datasets, we can also note that the original salient point methods failed to find the correct matches under huge viewpoint changes, but they all get expected performance levels by using the fully affine space framework and the proposed *K-order NNDR* matching strategy. Especially for the BRIEF, ORB, BinBoost and LATCH local descriptors which are easily influenced by scale, rotation and viewpoint changes, good performance was obtained for these changes by the framework of fully affine space and *K-order NNDR*.

Table 2.7: The comparison of computational cost and memory requirement in the framework of fully affine space.

Method	Tilt=5		Tilt=6		Tilt=7	
	Average numbers of points	Average memory requirement	Average numbers of points	Average memory requirement	Average numbers of points	Average memory requirement
Affine-SIFT+SIFT	55635	27.15M	65384	31.9M	74095	36.16M
Affine-SURF+SURF	79341	19.36M	99341	24.24M	119627	29.2M
Affine-SURF+RIFF	79341	21.74M	99341	27.22M	119627	32.77M
Affine-SURF+BRIEF	79341	2.38M	99341	2.98M	119627	3.58M
Affine-ORB+ORB	13314	0.4M	19263	0.58M	25805	0.77M
Affine-BRISK+BRISK	17565	1.05M	20583	1.24M	23066	1.38M
Affine-SURF+FREAK	79341	4.76M	99341	5.96M	119627	7.16M

2.6.3.3 Computational Cost and Memory Requirement

Computational cost and memory requirement are also important to the framework of fully affine space, because they reflect the computational complexity of the framework as well as the potential for the requirement of real-time systems. Considering that each salient point method extracts different amounts of local features in the fully affine space, we evaluated the average number of detected salient points and average memory requirement per image. The statistical results are summarized in Table 2.7.

It is worth noting that Affine-SIFT+SIFT and Affine-SURF+SURF consumed a huge amount of memory for the salient points detection and descriptor extraction in the fully affine space. For Affine-SIFT+SIFT and Affine-SURF+SURF, a large amount of salient points is extracted in the fully affine space framework and it increases the memory consumption correspondingly. We can also note that the memory requirement of binary string descriptors is less than that of real valued features. Moreover, as the performance show that the binary string features also achieved expected results under major viewpoint changes, integrating binary string features with *K-order NNDR* matching strategy in the framework of fully affine space is a good candidate for real-time systems.

2.7 Conclusions

In this chapter, we presented a comparison of detectors and descriptors on diverse image distortions and also evaluated their performance in the framework of fully affine space. According to the evaluation results, the FAST detector had the highest repeatability score compared to the score of other detectors, moreover it had the least detection time cost per point. Regarding the criterion of recall-precision, our experiments showed that the descriptors of SIFT, BRISK, and FREAK performed the best as affine invariant descriptors, and the time complexity showed that the binary descriptors provide very efficient feature description and matching.

In addition, for the special case of finding correspondences, we proposed the *K-order NNDR* matching strategy for the correspondences matching in the framework of fully affine space, and the experimental results show that the *K-order NNDR* is effective and obtained high accuracy correspondences under challenging image transformations. Furthermore, Affine-SIFT+SIFT showed the best performance on the correct correspondences in the framework fully affine space. When taking into account the computational complexity and memory requirement, binary string descriptors using the *K-order NNDR* matching strategy are a good trade-off between the accuracy and efficiency.

Chapter 3

RIFF: Retina-inspired Invariant Fast Feature Descriptor

In this chapter, we first present the Retina-inspired Invariant Fast Feature, RIFF, which is designed for invariance under scaling, rotation, and affine image deformations. The RIFF descriptor is based on the comparison of the intensity of pair-wise pixels over a sampling pattern that has similarities with the human retina. Then we introduce a strategy to improve accuracy by maximizing the discriminatory power of the point set. A performance evaluation with regard to Bag-of-Words based image retrieval on several well-known benchmark datasets demonstrates that the RIFF descriptor has competitive performance compared to state-of-the-art descriptors. Additionally, a popular approach from literature is to use visual words (or Bag-of-Words) constructed from real valued local descriptor (SIFT and SURF). To accommodate large scale data sets, we used an approximate nearest neighbor (ANN) based clustering approach to both real valued local descriptors and binary string local descriptors (BRIEF, ORB, BRISK, FREAK, Binboost and LATCH). The results on these test sets reveal that some of the recent binary string approaches outperform notable descriptors such as SIFT and SURF.

3.1 Introduction

Efficiently establishing the correspondences between images is very useful for numerous applications of computer vision, such as content-based image search, image classification, object tracking, and panorama stitching. Salient point methods are leading approaches, which have been proven to be effective in many real world applications.

In using salient points, one typically needs a detector and a descriptor. Detectors find the locations (e.g., blob, region, or point) in images which typically are in some way informative. The descriptor gives a model or representation of a local image region. Prior research of salient points has focused on high repeatability detectors and robustness under scaling and rotation [108].

The SIFT descriptor [14] is the most popular salient point method. It computes the Difference-of-Gaussian (DoG) operator in the Gaussian scale space, and assigns an orientation and a descriptor to each salient point based on the local gradient histogram. The SURF [12] salient point detector makes use of a box-filter to achieve efficient extrema detection in the scale space and it performs well with respect to the criterion of repeatability. The SURF descriptor of each detected salient point is calculated through summing Haar-wavelet responses in the defined region after orientation alignment. Recent binary string descriptors such as BRIEF, ORB, BRISK, and FREAK were proposed that have specific advantages such as low memory requirements as well as computationally efficient matching using the Hamming distance (bitwise XOR followed by a bit count). BRIEF [117] first uses Gaussian smoothing on the selected image patch, and creates a binary string descriptor by the comparison of the intensities of randomly sampled pixel-pairs around the patch center. ORB [118] employs the most efficient FAST [96] detector to determine the salient points in different layers of an image pyramid. It use the intensity centroid algorithm to determine the orientation for each point. The binary string descriptor of ORB is determined similar to BRIEF and effectively improves the robustness under image rotation and scale changes. BRISK [16] applies a FAST score as a measure to determine the extreme points in the image scale pyramid, and generates the descriptor by comparing pair-wise

intensities over a decreasing density circular sampling pattern. FREAK [17] also selects pairs of pixels over a decreasing density circular sampling pattern loosely inspired by the retina and then compares their intensities to form a binary vector. Both BRISK and FREAK use the sum of local gradients of selected pairs to estimate the orientation. Moreover, some local binary descriptors based on a supervised learning scheme also show good performance (BinBoost [123] and LATCH [124]).

The recently introduced salient point descriptors each have specific strengths. Some are most restrict to scale changes, whereas others are designed for speed and/or low memory requirements. Our goal was to design a descriptor which was optimized to be robust under affine image transformations including rotation and scaling. In this chapter, we first propose a novel discriminative salient point descriptor which is named "RIFF" because the sampling pattern is inspired by the distribution of cones (color vision) that can be observed in the human eye.

Moreover, empirical experiments conducted over the past decade have demonstrated that one of the most popular and successful approaches towards image similarity and visual concept analysis is to use salient point algorithms combined with visual word model and an approximate nearest neighbors (ANN) search [122]. This is mainly due to the robustness of salient point descriptors under various geometric transformations and to the introduction of the visual word model, which significantly improved the search efficiency and the adaptability to a particular image dataset. Current visual words systems are predominantly built using salient points algorithms such as SIFT and SURF whose descriptors are real valued. In contrast to the real valued descriptors, binary string descriptors were proposed in order to generate and use the feature descriptors in a more efficient way (e.g., BRIEF, ORB, BRISK, FREAK, BinBoost and LATCH). Another goal of this chapter is to give insights into the performance and requirements of these descriptors for large scale image search.

The main contributions of this chapter are as follows:

First, we proposed a salient point descriptor which outperforms current methods regarding robustness under affine image transformations. Moreover, we proposed

a measure to rank the generated salient point descriptors so that unstable points will be rejected and the discriminatory power of the set of descriptors will be improved. This is useful for speeding up the process of indexing and matching among a large amount of descriptors and increasing accuracy.

Second, we compared several of the most promising local descriptors on a wide variety of near duplicate transformations within the visual words paradigm. This is very important for computer vision applications because each social application may involve a different set of image transformations. Our results give some insight into which descriptors would be better or worse candidates in each of these cases. To our knowledge, this is the first contribution that compares visual word models generated by recent binary string features and applies on large scale image copy detection.

Third, we made a comparison of different types of features in terms of feature extraction and vocabulary generation by measuring, for example, computational efficiency as well as memory efficiency. This requirements are important because in some situations speed might be more important than accuracy alone. In addition we adopted the ANN search to achieve the vocabulary generation.

The rest of the chapter is organized as follows: In Section 3.2, we present the generation of our RIFF local feature descriptor. In Section 3.3, we describe the details of the visual word model generation. The datasets and evaluation criteria in the experiment are described in Section 3.4. The performance results of the proposed descriptor compared to current state-of-the-art descriptors are shown in Section 3.5, and finally conclusions are given in Section 3.6.

3.2 Discriminate RIFF Local Descriptor

3.2.1 Retina Sampling Pattern Review

The retina sampling pattern is based on the topology of the human retina as found in neuroscience research. This research reveals that the spatial distribution density of cone cells in the retina decreases exponentially with the increasing

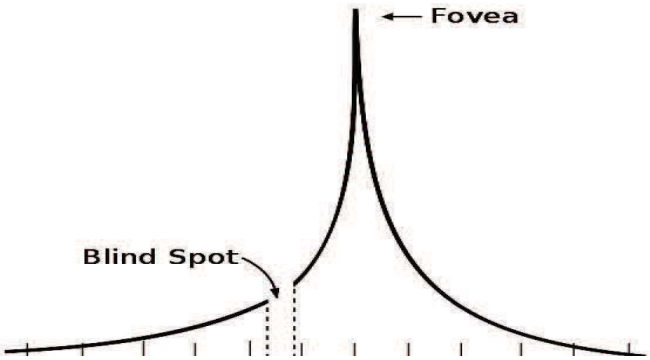


Figure 3.1: Illustration of the density distribution of cones in the human retina.

distance to the center of the fovea. Moreover, it is believed that the image signals pass through from cone cells to ganglion cells, where the receptive field of each ganglion cell uses the Difference of Gaussian (DoG) model with various sizes and that encodes differences into action potentials. Our approach employed a similar retina sampling pattern, which places different sizes of blocks at the defined locations in the pattern. The illustration of the cones density can be seen in Figure 3.1.

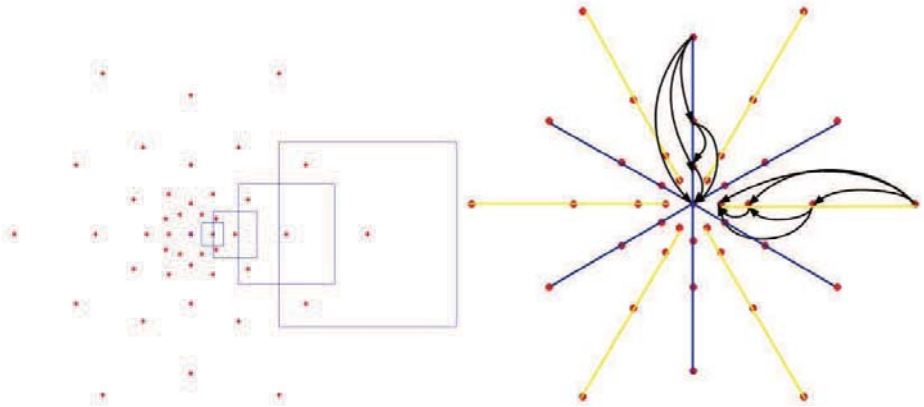
Inspired by recent work that use decreasing circular polar densities in diverse applications ranging from stereo matching to object recognition [16, 17, 125], the sampling pattern for RIFF in 2D decreases exponentially as shown in Figure 3.2 (a).

3.2.2 Descriptor Generation

3.2.2.1 Orientation Estimation

Given a set of salient points in an image (detected by the salient point detector), we first position and scale the retina sampling pattern according to the location and scale information (this is computed by the detector) for each specified point, and then calculate an orientation for them.

The popular approach for estimating the orientation angle comes from basic



(a) (b)

Figure 3.2: (a) The 2D exponentials decreasing polar sampling pattern for RIFF with N=43 points: the red points denote the sampling point locations, the blue rectangle represents a receptive field, and the size of the rectangle corresponds to its Gaussian kernel which is used to smooth the intensity values at the sampling points. (b) The pre-defined pair-wise point comparisons on RIFF for 2 of the 12 axes.

geometry which estimates the orientation using local gradients: $\triangle y$ and $\triangle x$ and then determine the angle from the arctangent of $(\triangle y/\triangle x)$ (for details see FREAK [17]). We also estimate the local gradients by pair-wise differences between equidistant points from the center of the retina sampling pattern.

3.2.2.2 Descriptor Generation

The procedure of RIFF descriptor generation is different from previous salient point approaches such as BRIEF, ORB, BRISK and FREAK, which compare the pixel-pair intensities in the sampling pattern to generate a binary string feature. Our approach first constructs a structure in the retina sampling pattern rotated by the estimated orientation θ. Let $V = [v_1, ..., v_i, ..., v_d]$ represent a feature vector of a salient point, where v_i is a real value obtained by calculating the difference of Gaussian smoothed image intensities of pre-defined pairs over the structure. We defined 6 pair-wise comparisons on each of the 12 axes from the center which results in the dimension of the descriptor d equal to 72. For clarity, we have

Figure 3.3: Matches (blue lines), between images after an affine viewpoint change, found by using the SIFT (OpenCV) salient point approach.

displayed in Figure 3.2 (b) 1 of the 6 pair-wise comparisons on the blue axes and 1 of the 6 pair-wise on the yellow axes where each black curve denotes one pair-wise comparison. Since we place a block at each sampling point, the integral image (summed area tables) was used for computational efficiency. It was not necessary for RIFF to compare the intensity of all possible $N \times (N-1)/2$ sampling pairs, which was necessary when calculating the binary string features used in previous methods. Moreover, the dimension of RIFF is smaller than SIFT, which may improve the speed of indexing and matching.

3.2.2.3 Discriminative Strategy

Even though location, scale and orientation have been estimated, current salient point detectors have difficulty with affine viewpoint changes such as depicted in Figure 3.3. We conducted a small internal study which revealed that local ambiguities (nearby salient points with similar feature descriptors) are often the cause of those matching errors.

Thus, our goal was to reduce local ambiguity or to increase local distinctiveness by eliminating salient points that have similar salient points nearby. We implemented this process by using a ranking scheme to identify stable local features. In this scheme, we consider a set of salient point descriptors $f_i, i = 1, 2, \ldots, M$, a salient point p in the image I and its feature f_p. The discriminatory score of the feature is defined according to the measure of similarity when compared to

K nearest neighbors in the image.

$$D_p(p \in I) = \sum_{j=1}^{K} \|f_p - f_i\|_2 \tag{3.1}$$

$\|.\|_2$ denotes the Euclidean distance. Intuitively, a higher discriminatory score demonstrates that the feature of point p is more distinctive than features of near by other points. The parameter K is set to 2 in the experiment, as it can achieve a good performance at a very low computational complexity. Furthermore, we use an exponential function in order to emphasize the discriminative score:

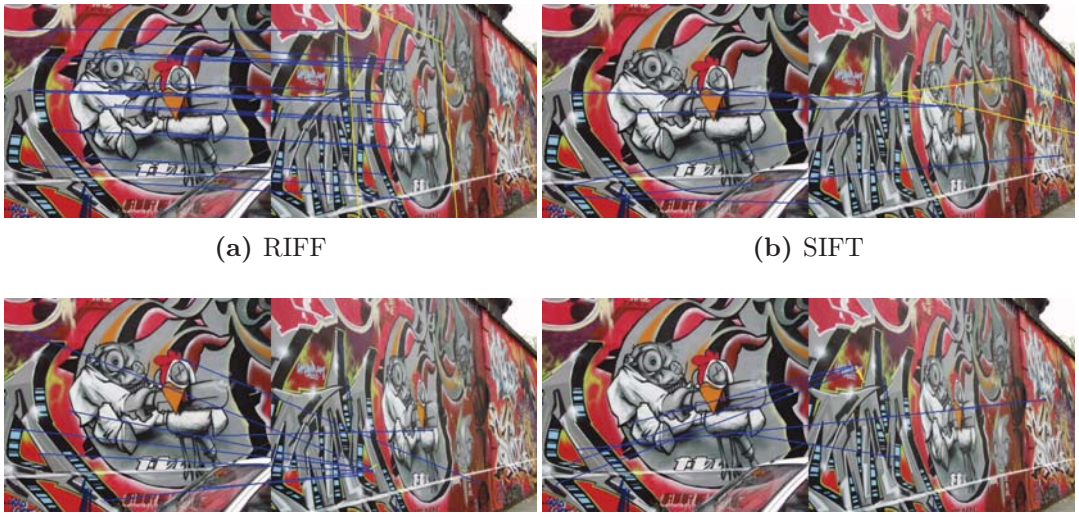
$$D_p^{'}(p \in I) = \exp(-\lambda \cdot |D_p|) \tag{3.2}$$

$|\cdot|$ denotes the normalization of D_p (in the range $[0, 1]$), λ is a weight of discriminative score and set to 6 that can achieve a good performance in the experiment. We note that after the above process, a smaller $D_p^{'}$ score correlates to more distinctive feature points, so we can sort these scores and define a threshold to filter out unstable salient points. The final set is a smaller number of discriminative features which are more robust to various image transformations, while reducing required subsequent processing, e.g., descriptor indexing as well as dictionary learning in large scale image applications.

We set the value of threshold in the *NNDR* to 0.75, and the homography between two compared images is estimated by the RANSAC algorithm. In preliminary tests, RIFF exhibited competitive performance for image copies detection under affine image transformations in comparison to the popular SIFT, SURF, and recent FREAK descriptors as shown in Figure 3.4.

3.3 Visual Word Model based Image Search

There are billions of images available on the WWW, scientific databases and private collections that do not have sufficient annotations for broad and accurate searching. Moreover, the number of images is ever increasing, and a large number



(a) RIFF **(b)** SIFT

(c) SURF **(d)** FREAK

Figure 3.4: Illustration of descriptor matching. Here RIFF is compared to SIFT, SURF, and FREAK on an image from a challenge on affine object detection (Graffiti 1-5 proposed by Mikolajczyk and Schmid [108]).

of similar copies exist. These copies can be viewed as transformed versions of the original images. Since common transformations such as geometric distortions, compression, crop, and color space changes could easily result in numerous copies or near-duplicates, it is a major challenge to achieve accurate, time and space efficient large scale detection of duplicates. Conventional global feature based image representations (color histogram, textual feature and shape information) can be used to perform an image search. However, they can not handle complex image transformations, such as rotation and scale changes. The visual word model based image representation (BoW [126], Fisher Vector [2] and VLAD [127]) takes advantage of the high discriminative capability of local descriptors in different contents and the applicability of different similarity measures to address complex image changes.

Visual word models, inspired by the field of information retrieval, were established by the introduction of salient point local descriptors, mainly because those local descriptors were shown to be invariant to scaling, rotation and noise. A visual word model represents an image as a histogram of visual words through feature

quantization and significantly improves the accuracy of image retrieval and object classification.

Typical implementations [4, 128, 129, 130] of the visual word model start by detecting salient points or regions from all images in the dataset and generate a descriptor for each salient point or region. These descriptors can be further clustered into a vocabulary consisting of visual words where each cluster center represents a visual word, and the size of the vocabulary is equal to the number of clusters. Based on salient points extracted as salient image patches, an image is frequently represented using a histogram according to the occurrence frequency of each visual word.

For the popular real valued local descriptors (e.g., SIFT, SURF), the simple K-means clustering algorithm can be used to train the visual word vocabulary. The initialization of cluster centers is first generated by randomly choosing candidates from the descriptors group. After that, at the beginning of each iteration, the remaining descriptors are assigned to their closest cluster center. The center can be updated by the mean value of the assigned descriptors. Euclidean distance is used as a distance measure in the assignment procedures.

For binary string descriptors, the Hamming distance metric is used. As it only use bitwise XOR followed by a bit count, it offers a higher matching speed. As the traditional computation of an average is not suitable for binary features, we employed an approach named "K-majority" [131] to calculate the mean value of binary string descriptors.

The K-majority method refines cluster centers based on the statistics of the total number of 1's at the same bit position among all the descriptors belonging to the same cluster. Suppose a cluster consist of I binary string features: $F_i, i = 1, 2, ..., I$, and we treat a binary feature as $F = [bit_1, bit_2, ..., bit_J]$, $1 \leq j \leq J$, where J denotes the length of binary string feature. The following function can then be used to update the cluster center.

$$score(bit_j) = \sum_{i=1}^{I} F_i(bit_j)/I \tag{3.3}$$

$$Center(bit_j) = \begin{cases} 1, & if : score(bit_j) \geq 0.5 \\ 0, & if : score(bit_j) < 0.5 \end{cases} \tag{3.4}$$

Function (3.4) implies that if the number of 1's is larger than half the number of total descriptors belonging to the specified cluster, the new value of the same bit position of the center is set to "1", otherwise it is set to "0".

However, it is a challenge to apply the flat K-means or flat K-majority to large scale vocabulary construction, because it is computationally expensive to perform clustering in high dimensional spaces. In order to reduce the computational complexity of linear search, an approximate nearest neighbors approach (ANN) was adopted to assign the labels of optimal cluster centers to descriptors. Compared with the flat K-means and flat K-majority, ANN-based K-means and K-majority approaches could effectively reduce the complexity from $O(NK)$ to $O(Nlog(K))$ during each iteration, where N denotes the number of descriptors, and K is equal to the number of centers. Considering the different properties of real valued descriptors and binary string descriptors, ANN search is based on a KD-tree index and a LSH index respectively [132]. The LSH index space is based on multi-probe LSH, which has the advantage of reduced storage requirements. Once the visual vocabulary has been obtained, we represent an image as a bag of visual words according to the popular *tf-idf* weighting scheme [133]. The *tf-idf* weighting scheme can reduce the contributions of common visual words, while at the same time increasing the contributions of discriminative words. Through building an index for the image features in the dataset, a ranked list of search results could be efficiently returned according to the distance similarity with the query image feature.

3.4 Experimental Results

The experimental environment for the evaluation is an Intel Quad Core i7 Processor (2.67GHz), 12GB of RAM, 64-bit OS. The implementations of BinBoost is from the author, others are implementations from OpenCV. The parameters of

each salient point method were set to the defaults. We used 8 randomized forests in the KD-tree index, 20 hash tables in the multi-probe LSH index. Our evaluation implementations are available at: http://press.liacs.nl/researchdownloads/.

3.4.1 Datasets and Evaluation Criteria

The evaluation of visual word based large scale image copy detection is performed on three image datasets: PASCAL VOC2012 [134], Caltech 256 [135], and MIR FLICKER 1Million [136]. Moreover, a series of near duplicates were created for the test. We use mAP (mean Average Precision) as a criterion for the evaluation of detection accuracy. The transformed duplicates categories generated for the test mainly include: cropping, content noise, image blur, image compression: JPEG compression, rotation, scale and affine deformation: rotation + scale + 3D perspective distortion.

Scale change: we resized the original images by changing the scale factors from 20% to 200% with a step size of 20%.
Cropping: starting with a 50×50 pixel central region in the image, the width and height of the cropped area of the image is gradually increased by 10 pixels.
Image compression: JPEG compression copies are produced by setting the image quality factors in the range from 95% to 5%.
Text noise: images are modified by adding various sizes and colors of text in the central area.
Image blur: A series of blurred images is created by smoothing the image using Gaussian smoothing.
Deformation: includes several subsets where image copies (rotation, rotation together with scale, and viewpoint transformation) are created by rotation as well as perspective distortion with different angles.

A total number of 1000 images in each dataset are randomly selected as query images, and 80 duplicates of each query image are generated. Some examples of each dataset used for evaluation are illustrated in Figure 3.5.

(a)

(b)

Figure 3.5: Examples from each dataset for the evaluation of salient point methods. (a) Examples from the VOC, Caltech 256 and MIR datasets. (b) Examples of generated duplicates: text noise, JPEG compression, image blur, image crop, rotation and affine transformation, respectively.

3.4.2 Evaluation of Image Copy Detection

In this section, we focus on constructing the visual word vocabulary not only by using real valued descriptors, but also binary string descriptors. We use ANN search to efficiently train the vocabulary. We first compared the proposed RIFF with a number of the most promising salient feature descriptors on a wide variety of near duplicate transformations within the visual words paradigm. This is the most important part of this section because each application may involve different image transformations and our results give some insight into which descriptors would be better or worse candidates. Then, we made a comparison of different types of features in terms of feature extraction and vocabulary generation by measuring indicators of computational efficiency as well as space requirements. There characteristics are valuable because in some situations speed, for example, might be more important than accuracy alone.

In order to make an objective comparison of different types of local descriptors, we also choose to use the same detector for each local descriptor. SURF was applied as the salient point detector, and we combine the SURF detector with various fea-

ture descriptors including SIFT, SURF, ORB, BRIEF, BRISK, FREAK, RIFF, BinBoost and LATCH in the evaluation. The performance of various vocabularies is evaluated in terms of computational efficiency, memory requirements, and accuracy.

The criterion to estimate the similarity of two images represented by visual words is via the cosine distance measure. We use mAP (mean Average Precision) as a criterion to evaluate the performance of search accuracy.

3.4.2.1 Evaluation of Time and Storage

We first focus on the efficiency and space requirements of generating the vocabularies for the different descriptor types. For this evaluation we use the PASCAL VOC dataset. Ten million salient point features were extracted from the dataset.

Figure 3.6 illustrates the computational efficiency of different types of vocabulary generation as well as the storage space requirement under different cluster sizes. The vocabulary generation based on the compared descriptors all reveal an almost linear growth with increasing vocabulary size. In Figure 3.6 we can see that the execution time of the vocabulary training stage with real valued descriptors is nearly 4 times faster than that of binary string features, however, binary type vocabularies have significantly lower space requirements.

3.4.2.2 Evaluation of Search Accuracy

We evaluated the performance of image copy detection using various visual vocabularies. As all the generated duplicates are added into the datasets, the scale of PASCAL and Caltech is roughly 10 thousand, and MIRFLICKR contains around one million. The comparison experiment with different types of vocabulary is based on varying the vocabulary size.

Overall, the RIFF based visual words model outperformed the other descriptors on the PASCAL VOC, Caltech 256 and MIRFLICKR-1M datasets as shown in

Figure 3.7. The mAP score results also demonstrate that binary local descriptor based visual vocabularies offer good performance. Comparing the evaluation results on the one million-scale dataset and the results on VOC dataset, there is no significant mAP score decrease when the data size increased from ten thousand to more than one million. We can also note that the FREAK descriptor based vocabulary has better mAP score on average across the three datasets than other binary string based vocabularies. Below we will discuss how various descriptor based visual word models performed under the different transformations.

Our goal of this part is to determine the robustness of the visual vocabularies to various image transformations. As shown in Figure 3.8, RIFF had the best performance on the distortions related to scale, rotation, and affine transformations. It showed average performance on blurring and showed competitive performance on the rest of the transformations. When the transformation keeps the structure in place such as blur and JPEG compression, SIFT has high accuracy but was outperformed by BRIEF, while BinBoost showed a weakness for the cases of blur and JPEG compression. We observe that when pictorial information is added to or deleted from an image copy, SIFT was consistently outperformed by the other descriptors. Specifically, FREAK performed well on transformations which deformed the image structure such as affine transformations or combining rotation with scaling. BRIEF showed particularly poor performance on rota-

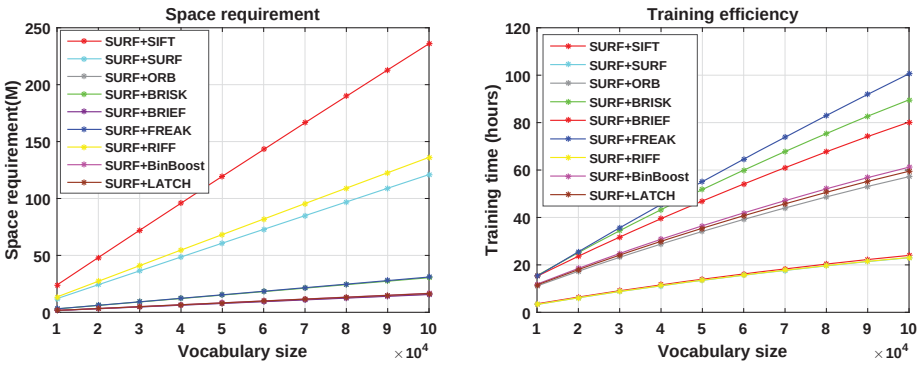


Figure 3.6: Comparison of different descriptors in terms of time efficiency and space requirements during the training. Both space requirement and training time show almost linear growth when the size of the vocabulary increases.

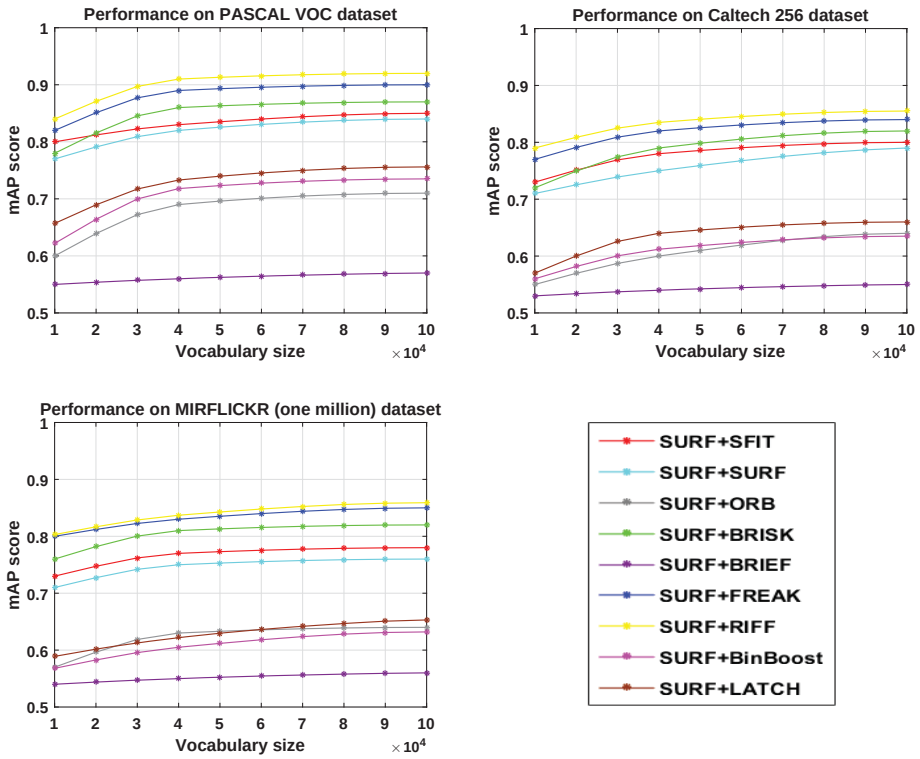


Figure 3.7: Detection accuracy (mAP score) on three datasets (PASCAL VOC, Caltech 256 and MIRFLICKR). The size of the vocabulary varies from 10000 to 100000.

tion transformations and LATCH showed a poor performance on scale changes. Note that the affine deformation represents the most difficult category, as the total number of detected copies is extremely low for all types of compared visual vocabularies.

According to the copy detection accuracy, the robustness of visual word model based image representations mainly rely on the capability of the local descriptor. We can see that the BRIEF descriptor is not rotation and scale invariant, thus, a visual word model trained on BRIEF is sensitive to rotation and scale changes. The ORB descriptor makes an improvement in case of the rotation changes when compared to the BRIEF descriptor, therefor, vocabularies trained on the ORB descriptors showed better performance than BRIEF. RIFF, BRISK and FREAK based visual word models have high performance for rotation and scale invari-

ance, probably because the local descriptors of RIFF, BRISK and FREAK use circular sampling patterns. The vocabularies trained on new binary string features (BRISK and FREAK) and the real valued features (SIFT, SURF and RIFF) all are scale invariant and robust to JPEG compression and blur noise. For the category of learning based local descriptors, the learning scheme of BinBoost is not robust to the JPEG compression and blur noise. LATCH does not use scale information during the learning process.

3.5 Conclusions

We have proposed a novel salient point descriptor named RIFF which was inspired by the sampling pattern used by the human eye (we make no claims of biological relevance). The main contribution of the RIFF descriptor is in constructing the descriptor so that the discriminatory power is optimized by ranking and deleting points with low distinctiveness. Our Bag-of-Words image retrieval tests on three well known datasets, showed RIFF outperforming the other feature descriptors with respect to robustness to scale, rotation, and affine transformations. Furthermore, we presented a performance evaluation of real valued and binary string salient point descriptors. The time complexity and space requirements showed that binary string descriptors are efficient in terms of feature extraction time and memory usage. Regarding the criterion of the mAP score, the image copy detection experiments showed some significant strength of binary string local descriptors. FREAK clearly outperformed SIFT on rotation and scale, and affine transformations. BRIEF had the best accuracy in case of image blur and was among the best in case of image cropping.

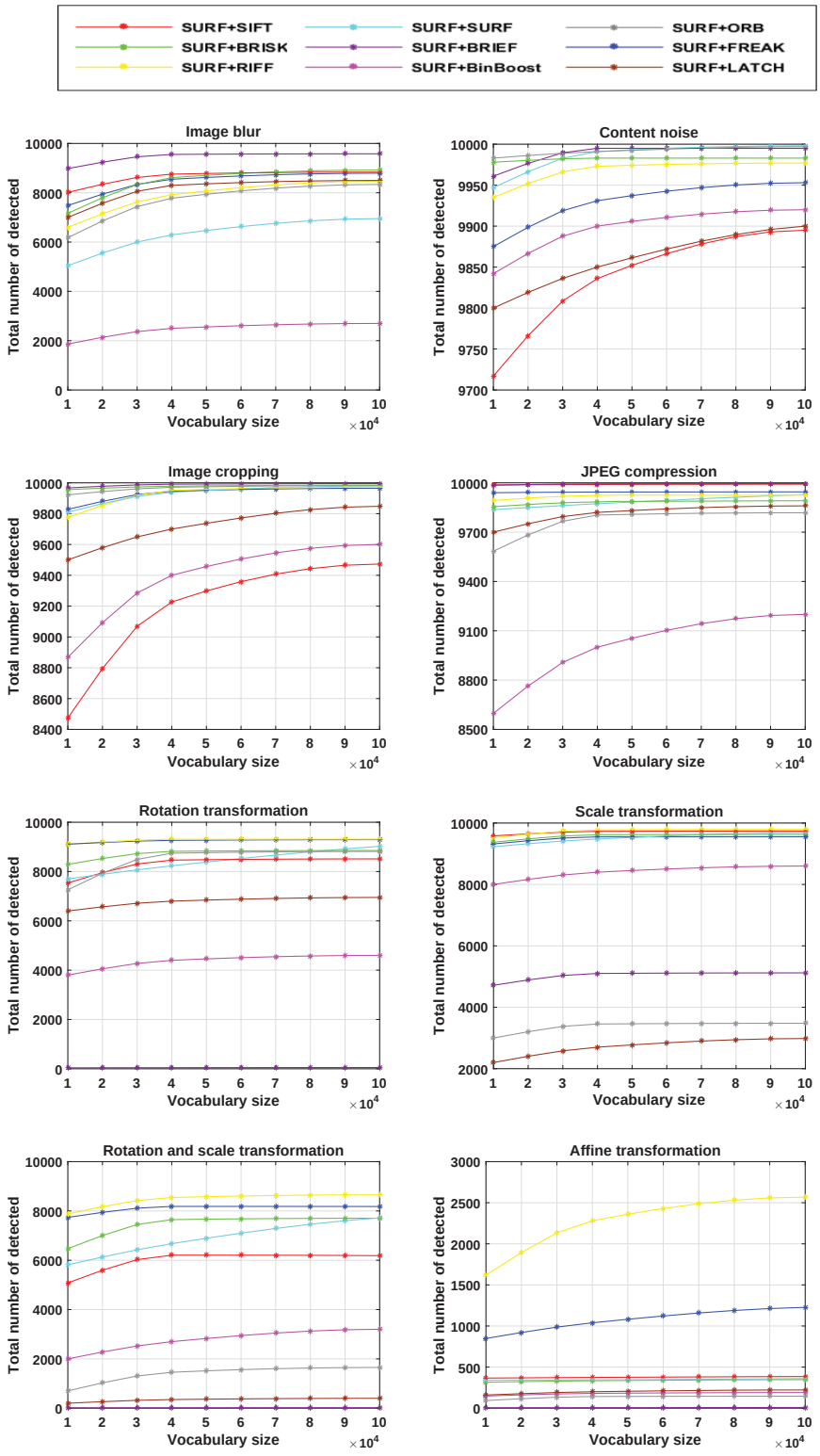


Figure 3.8: Total number of detected duplicates from different types and different sizes based vocabularies in each transformation category. The size of vocabulary varies from 10000 to 100000.

Chapter 4

Deep Binary Codes for Large Scale Image Retrieval

Recent studies have shown that image representations built upon deep convolutional layers in Convolutional Neural Networks (CNNs) have strong discriminative characteristics. In this chapter, we present a novel and effective method to create compact binary codes (deep binary codes) based on deep convolutional features for image retrieval. Deep binary codes are generated by comparing the response from each feature map and the average response across all the feature maps on the deep convolutional layers. Additionally, a spatial cross-summing strategy is proposed to directly generate bit-scalable binary codes. As the deep binary codes on different deep layers can be obtained by passing the image through the CNN and each of them makes a different contribution to the search accuracy, we then present a dynamic, on-the-fly late fusion approach where the top N high quality search scores from deep binary codes are automatically determined online and fused to further enhance the retrieval precision. Two strengths of the proposed methods are that the generation of deep binary codes is based on a generic model, which does not require additional training for new image domains, and that the dynamic late fusion scheme is query adaptive. Extensive experimental results on well known benchmarks show that the performance of deep binary codes are competitive with state-of-the-art approaches for large scale image retrieval. Moreover,

it is shown that the dynamic late fusion scheme substantially enhances the search accuracy.

4.1 Introduction

Content-based image retrieval aims to find relevant images in an image database that share a similar appearance with a given query image. This is a challenging task for large scale visual search, because one must address both the typical appearance transformations such as changes in perspective, rotation and scale; and also minimize memory, computational cost, and response time.

Traditional image retrieval systems based on visual word representations mainly owe their success to locally invariant features and large visual codebooks. The Bag-of-Words (BoW) [1] approach is usually employed to encode local features into a histogram according to the occurrence frequency of each visual word. Peronnin *et al.* proposed the Fisher Vector [2]. The visual words in a Fisher Vector are constructed with a Gaussian mixture model (GMM) where the gradients of local features corresponding to particular parameters in GMM are summed. The Fisher Vector image representation is the concatenation of each accumulated gradient. Jegou et al. proposed a Vector of Locally Aggregated Descriptors (VLAD) [3] to capture more information from the image. VLAD and its variations [4, 5, 6] are viewed as a type of simplified Fisher Vector, and it accumulates the difference of each local feature to the visual words and concatenates these accumulated values to describe an image.

Visual word based approaches are challenging to scale to very large image databases, as they have significant computational and memory requirements. Hashing techniques, such as iterative quantization (ITQ) [137], locality-sensitive hashing (LSH) [138], spectral hashing (SH) [23], spherical hashing (SpH) [139], locality-sensitive hashing from shift-invariant kernels (SKLSH) [20], density sensitive hashing (DSH) [140] as well as PCA-random rotation (PCA-RR) [137] focus on learning compact yet powerful image representations for efficient large scale visual search. The basic idea of hashing-based approaches is to construct a hash function to

map each visual object into a binary string code such that similar visual objects are mapped into similar binary codes. Unlike the above mentioned hashing approaches, which seek a linear function to project data into a binary vector, recent supervised hashing methods based on convolutional neural network (CNN) architectures [141, 142] seek to learn multiple hierarchical non-linear transformations to generate distinctive binary codes. However, most state-of-the-art hash function learning methods require additional training for each new image domain. This can require significant resources both for assembling the supervised training data and the learning process.

The recent CNN based image representation makes use of the transfer property of a CNN architecture that is pre-trained on a large scale dataset. It has been shown to provide a highly discriminative descriptor representing an image and to produce superior performance in various computer vision tasks, such as image classification, object detection and visual search [58, 143, 144, 145, 146]. Most of these research projects utilize the outputs from the fully connected layers to represent images (directly used or followed by PCA reduction [63]). In particular, visual representations from activations of deep convolutional layers have been shown to lead to high accuracy for image retrieval in real world image test sets. This is achieved by processing a max-pooling, spatial max-pooling [64, 147] or sum-pooling [65] operation on the deep convolutional layers. Better performance is obtained using deep convolutional features than if the features from the fully connected layers are used. This is mainly due to the fact that the activations in each channel of the convolutional layer correspond to receptive fields in the original image, i.e., having a direct semantic interpretation.

Inspired by the advantages of image representation through aggregating activations from deep convolutional layers, we propose a novel and efficient approach to construct bit-scalable binary codes from deep convolutional layers for highly efficient image retrieval (as shown in Figure 4.1). This idea is mainly based on the fact that similar visual objects have similar distributions of responses of feature maps on deep convolutional layers. In this chapter, we propose to generate the binary code on each convolutional layer according to the comparison between the response of each feature map and the average response across all the feature

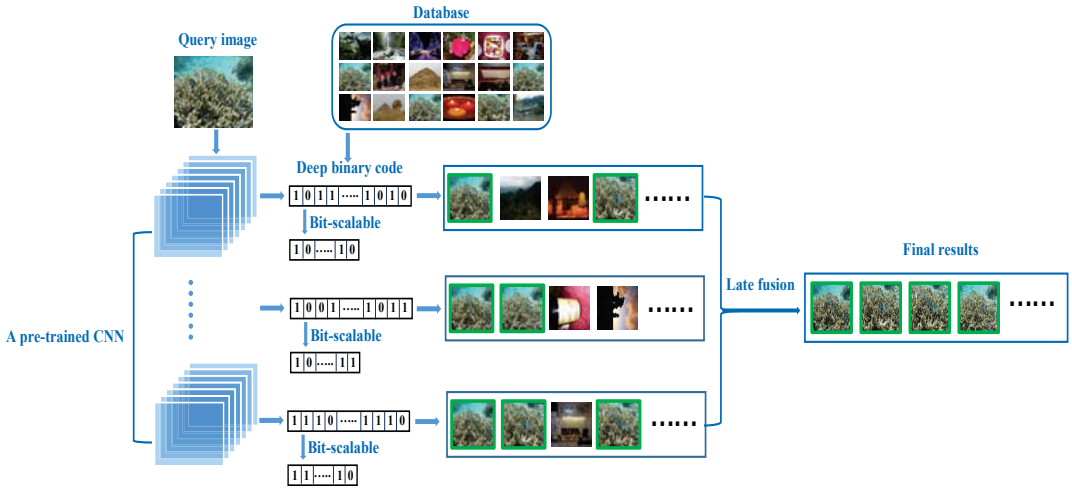


Figure 4.1: The proposed image retrieval framework. Our method consists of two main components. The first is the deep binary code generation on each deep convolutional layer of a pre-trained CNN. In the second component, we propose a dynamic late fusion scheme to further increase the search precision. Images with green rectangles are positive results.

maps on the same deep convolutional layer. Additionally, a strategy of spatial cross-summing is designed to generate bit-scalable deep binary codes. Extensive experiments on well-known image retrieval benchmarks demonstrate the effectiveness of the proposed binary code representation (referred to as deep binary codes) and show competitive results compared to state-of-the-art image retrieval approaches.

The strengths of the proposed deep binary codes are three-fold. First, the deep binary code is highly efficient regarding computational and memory costs. By passing a test image through a pre-trained CNN architecture, the compact binary codes on each deep convolutional layer can easily be generated. Second, the length of a deep binary code can be controlled by the spatial cross-summing operation. Third, available pre-trained CNN architectures (VGGNet [51], AlexNet [48] as well as GoogleNet [42]) can be directly employed to generate deep binary codes.

It is worth to note that during the procedure of passing an image through a

pre-trained CNN architecture, all the deep binary codes from lower to higher layers can be obtained. The similarity scores given by deep binary codes from different layers vary largely. As illustrated in Figure 4.3, for a specific query image, the average precision score of each deep binary code is different, and it is difficult to determine in advance which deep binary code is the most robust one. Thus, we are motivated to investigate how to fuse the search scores returned by deep binary codes from different layers, to further improve the precision of visual search. Inspired by the idea proposed by Zhang et al. [148] which demonstrates that the score curve returned by a good feature shows an "L" shape, while that returned by a bad feature shows a gradually dropping tendency, the effectiveness of a feature can be estimated, as it is negatively related to the size of the area under the normalized and sorted score curve. In this chapter, we propose to optimize the operation of normalization in Zhang et al.'s method and design a new unsupervised dynamic late fusion scheme to choose the top N good features for a given query, and then aggregate the search scores of the top N candidates to improve the search precision.

The main contributions of this chapter are summarized as follows:

First, this chapter introduces a novel and compact deep binary representation which is generated from the convolutional layers of pre-trained CNN architectures and investigates the reasons underlying its success. The proposed approach creates bit-scalable deep binary codes in a data-independent manner in the sense that it uses a generic transferred model, which does not require additional training.

Second, image representations based on different pooling operations (such as max-pooling, average-pooling and sum-pooling) as well as various hashing function learning methods on the activations of the deep convolutional layers are evaluated. This results in both insights and a baseline for large scale image retrieval.

Third, the proposed adaptive and unsupervised dynamic (top N) score-level late fusion scheme is shown to significantly improve the image retrieval accuracy.

The remainder of this chapter is organized as follows. First, we briefly review related work in Section 4.2. Section 4.3 introduces the details of the proposed deep

binary codes and the dynamic score-level late fusion scheme. The experimental results are presented in Section 4.4 and conclusions are given in Section 4.5.

4.2 Related Work

CNN based image representation: deep learning aims to learn higher semantic representations by passing an image into the architecture of a convolutional neural network [48]. The image features generated from the activations of the fully connected layers have already demonstrated their good performance in image retrieval tasks. Recent research projects further explore the features from the deep convolutional layers. Ng et al. [149] treated the channels from one deep convolutional layer as visual words and encoded them into a feature similar to a VLAD. Razavian et al. [64] and Azizpour et al. [147] proposed to aggregate the activations from the last convolutional layers by max-pooling or spatial max-pooling which show better performance than those from fully connected layers. Additionally, it was revealed that the image representation by sum-pooling and PCA whitened on the last convolutional layer leads to much better performance [65]. Giorgos et al. [150] proposed to extract a set of features by max-pooling at multiple scales on a deep convolutional layer and subsequently summing the collected features to describe an image.

Learning based hashing: the existing hash function learning methods can be classified into two categories: data-independent and data-dependent. LSH [138] is a representative data-independent method which proposed to use random projections to map data into binary codes. SKLSH [20] is an extension of LSH which extends Euclidean distance to other distances. For the data-dependent categories, the method of SH [23] was presented to obtain balanced binary codes by solving a spectral graph partitioning problem. ITQ [137] creates binary codes by simultaneously maximizing the variance of each bit and minimizing the quantization error. SpH [139] was proposed to preserve the data locality relationship to keep neighbors in the input space as neighbors in the Hamming space. CNN based hashing methods with supervisory information in the form of class labels have

been further developed by Lai et al. [141] and Zhao et al. [142], which optimize the CNN architecture based on a loss function to preserve binary semantic similarity of the data. Compared with the hashing-based learning methods, the proposed deep binary codes achieved competitive and even better performance without requiring training data and supervisory information.

Late fusion approaches: Late fusion approaches fuse the search results from different features or different methods to increase the search accuracy. Nandakumar et al. [151] proposed a framework which optimally combines the genuine match scores through the likelihood ratio calculation. Zhang et al. [152] proposed a graph-based query specific fusion method where multiple retrieval lists obtained by different methods are merged and re-ranked by a graph model. Zheng et al. [148] proposed to determine the weight of different search scores based on the fact that the quality of a feature has a negative relationship to the area under the normalized score curve. Our proposed method is similar to [148], however the difference is that our late fusion approach only combines the search scores from the top N high quality features, without requiring expensive offline calculations for different features.

4.3 Proposed Approach

4.3.1 Generating Deep Binary Codes

In this section, we describe how to generate the deep binary codes. This starts with a pooling operation, which calculates a summary statistic (such as max-pooling, sum-pooling, multi-scale-max-pooling or multi-scale-sum-pooling) over a local spatial region on the deep convolutional layers. The main motivation behind the use of pooling is to promote invariance to local input transformations (such as translation, occlusion and truncation of the local stimulus), which could greatly improve the effectiveness of the deep convolutional layer representation. This is due to fact that the resulting outputs by pooling are invariant to their spatial location within the pooling region. This is particularly important for the

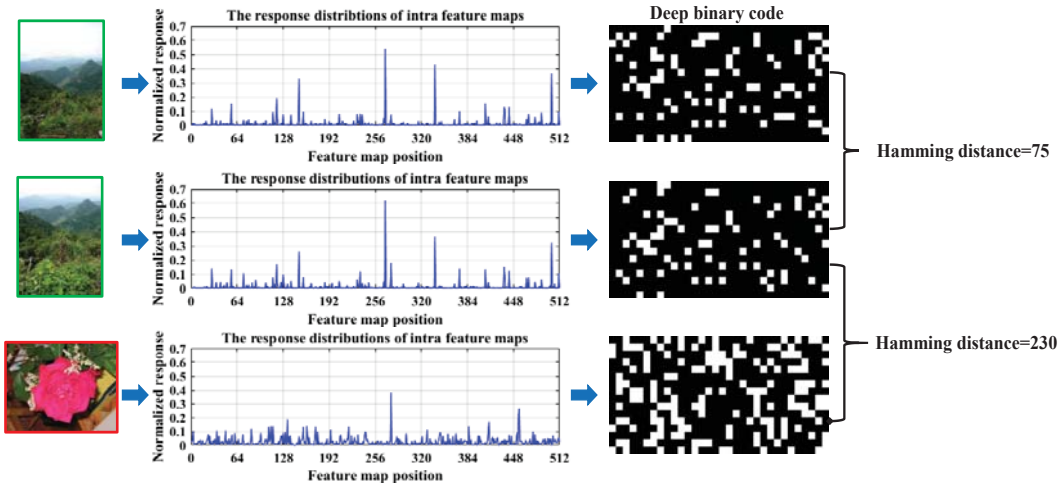


Figure 4.2: The intra feature map distribution between relevant images and irrelevant images. Relevant images have similar distributions and the peak values appear at the same positions, which results in similar deep binary codes.

performance of image search where local image transformations obfuscate object identity. An additional advantage of the pooling operation is that it reduces the spatial resolution, resulting in a lower-dimensional image representation.

Consider a pre-trained CNN architecture with L deep convolutional layers, and a given input image I. We pass I through the pre-trained network where the obtained feature maps can be denoted as $\bar{F}_i = \{F_{i,j}\}$ with $i = 1...L, j = 1...C_i$, where the $F_{i,j}$ is equal to the j^{th} feature map at the i^{th} deep convolutional layer and C_i is equal to the number of channels (or convolutional kernels) of deep layer i. Assume that $F_{i,j}$ has size $W_i \times H_i \times C_i$, where W_i and H_i are the width and height of each channel, respectively. We further associate each cell in the feature map from the i^{th} layer with a spatial coordinate (x, y) and the response at this position $f_i(x, y)$. Then, the image representation by max-pooling on a deep convolutional layer can be described as follows:

$$\bar{\mathbf{V}}_\mathbf{i} = [\bar{V}_{F_{i,1}}...\bar{V}_{F_{i,j}}...\bar{V}_{F_{i,C_i}}] where\ \bar{V}_{F_{i,j}} = \max_{x,y \in F_{i,j}} (f_i(x, y)) \tag{4.1}$$

The max-pooling operation encodes the local maximum response from each feature map and leads to a compact feature vector with its dimension equal to the

number of feature maps.

In contrast to max-pooling which only makes use of the local maximum response in the feature map, sum-pooling encodes all the responses into the feature vector. Sum-pooling on activations from a deep convolutional layer can be calculated as follows:

$$\hat{\mathbf{V}}_\mathbf{i} = [\hat{V}_{F_{i,1}}...\hat{V}_{F_{i,j}}...\hat{V}_{F_{i,C_i}}],\ \hat{V}_{F_{i,j}} = \sum_{x=1}^{H_i}\sum_{y=1}^{W_i} f_i(x,y) \tag{4.2}$$

As the activations from the convolutional layers can be interpreted as local features corresponding to particular original image regions, the simple max-pooling and sum-pooling do not consider the spatial and location information of the activations in the feature map, hence the generated feature vectors are only translation invariant. Furthermore, as the local regions appear at various scales in the images, the scheme of multi-scale-pooling on feature maps is utilized to capture information at different scales. In this way the image representation could be robust to scale transformations. Let R denote a region in a feature map, then the extracted feature in this region by pooling can be constructed as follows:

$$\mathbf{V}_{\mathbf{i},\mathbf{R}} = [V_{F_{i,1,R}}...V_{F_{i,j,R}}...V_{F_{i,C_i,R}}], V_{F_{i,j,R}} = \underset{x,y\in R}{P} \mid f_i(x,y) \mid \tag{4.3}$$

$$\mathbf{V}_\mathbf{i} = [V_{F_{i,1}}...V_{F_{i,j}}...V_{F_{i,C_i}}],\ V_{F_{i,j}} = \sum_{R\in F_{i,j}} \mathbf{V}_{\mathbf{i},\mathbf{R}} \tag{4.4}$$

The function $P \mid \cdot \mid$ can be max-pooling, sum-pooling or average-pooling on the region R. R is a square region of the feature map with width (height) from 1 to $\min(W_i, H_i)$. The extracted features from multiple scale regions are then summed, and subsequently l_2-normalized to represent the image.

We further observe that: for a pair of similar images, the feature maps with a high response appear at almost the same index positions on the deep layer (referred to as intra feature map distribution), as shown in Figure 4.2. Based on this observation, we propose to convert the image representation on the deep convolutional layers into binary codes $\mathbf{B}_\mathbf{i} = [B_{F_{i,1}}...B_{F_{i,j}}...B_{F_{i,C_i}}]$. This binary

code is constructed by comparing the response from each feature map with the average response across all the feature maps:

$$B_{F_{i,j}} = \begin{cases} 1, \ if \ V_{F_{i,j}} \geq average(\mathbf{V_i}) \\ 0, \ if \ V_{F_{i,j}} < average(\mathbf{V_i}) \end{cases} \tag{4.5}$$

Thus, the image representation by the deep convolutional layers is converted into binary codes in an unsupervised and training data independent way. Moreover, these binary codes have low memory requirements and allow for fast matching using the Hamming distance, hence binary codes are very suitable for large scale image search.

4.3.2 Spatial Cross-Summing

Based on the pre-trained CNN architecture, the bit-length of the deep binary codes is preset according to the number of feature maps in the deep layers. In real-world applications, binary codes with different bit-lengths allow researchers to make trade-offs between accuracy and efficiency. For example, real-time systems and devices with limited computational and storage resources require low dimensional binary codes, while higher dimensional binary codes are more appropriate for increased accuracy. The conventional PCA-operation is data-dependent and is not suitable for the generation of bit-scalable deep binary codes. To address these issues, we propose a spatial cross-summing strategy to create compact and bit-scalable deep binary codes from deep-layer features.

For a given deep-layer feature $\mathbf{V_i}$ with length C_i, the objective is to generate a bit-scalable deep binary code with length n, $n = C_i/2^m$, and $m = 1, \ldots, \log_2(C_i)$. This procedure starts by generating the deep-layer feature with length n by a spatial cross-summing strategy. For example, let $n = C_i/4$, then $\mathbf{V_i^{2n}} = \mathbf{V_i}[1, 2, ..., 2n] + \mathbf{V_i}[C_i, C_i - 1, ..., 2n + 1]$, and $\mathbf{V_i^n} = \mathbf{V_i^{2n}}[1, 2, ..., n] + \mathbf{V_i^{2n}}[2n, 2n - 1, ..., n + 1]$. Finally, the deep binary code $\mathbf{B_i}$ is calculated using Formula (4.5) on the vector $\mathbf{V_i^n}$. Algorithm 1 formalizes the procedure of bit-scalable deep binary code generation.

Algorithm 1 : Bit-scalable deep binary codes generation

Input: the i^{th} deep-layer image feature: $\mathbf{V_i}$ with length C_i, and $n = C_i/2^m$, gives $m \in N \geq 1$

Output: deep binary codes $\mathbf{B}$ with n bits

$X \Leftarrow C_i/2$, $\mathbf{V} \Leftarrow \mathbf{V_i}$
while $X \neq n$ **do**
  $bit \Leftarrow X$, $l \Leftarrow length(V)$
  $\mathbf{V^a} \Leftarrow [V_1, V_2, \cdots, V_{bit}]$, $\mathbf{V^b} \Leftarrow [V_l, V_{l-1}, \cdots V_{l-bit+1}]$
  $\mathbf{V'} = \mathbf{V^a} + \mathbf{V^b}$
  $X \Leftarrow X/2$, $\mathbf{V} \Leftarrow \mathbf{V'}$
end while
Deep binary code $\mathbf{B}$ generation of size n bits
for all V_{bit} in $\mathbf{V}$ and B_{bit} in $\mathbf{B}$ **do**
  if $V_{bit} \geq average(\mathbf{V})$ **then**
    $B_{bit} \Leftarrow 1$
  else
    $B_{bit} \Leftarrow 0$
  end if
end for
return $\mathbf{B}$ of size n bits

4.3.3 Dynamic Late Fusion

Another advantage of the proposed binary string representation is that the deep binary codes on different layers could all be generated by passing the input image through the pre-trained CNN just once, without additional re-feeding operations. Moreover, different deep binary codes make different contributions to the image search. As illustrated in Figure 4.3, the deep binary code from the conv5 layer gives a higher average precision score than that from the pool5 layer. Thus, the critical issue is how to automatically measure and compare the quality of each deep binary code, since no supervision and relevance feedback are available online, and the only accessible information is the search scores returned by different deep binary codes. Therefore, we aim to exploit these search scores to improve the

Figure 4.3: The deep binary codes from pool3, conv5 and pool5 are employed to obtain three sorted search scores respectively, where the code from conv5 produces good performance (AP =0.95) and has a smaller area under the curve than those from pool3 and pool5. Note that the curve from pool3 goes beneath that from pool5 after the marked yellow line (which needs to be avoided).

retrieval performance.

The authors of [148] show that the curve of a sorted search score returned by a good feature appears to have an "L" shape and the curve returned by a bad feature shows a gradually decreasing tendency. Furthermore, they showed that the size of the area under the sorted score curve can be used as an indicator to identify the quality of the features. Motivated by this, we fuse the search scores from the top N good deep binary codes.

For a specific query image I, together with a set of deep binary codes $\mathbf{B_i}$, $i = 1...L$, we can use the Hamming distance to measure similarity. Note that in case of the Hamming distance higher similarity corresponds to a lower value. We use a modified Hamming distance $\bar{H}_I = K - H_I$ such that $\bar{H}_I$ has a higher value for higher similarity. Here K is the size of the deep binary code and the sorted search score based on the modified Hamming distance returned by one deep binary code is represented by S_i. We further use max-min normalization on the sorted search scores returned by the modified Hamming distance, so that relevant images for a

query give a max score equal to 1, while irrelevant images give a score of 0.

$$\bar{S}_i = \frac{S_i - min(S_i)}{max(S_i) - min(S_i)} \tag{4.6}$$

The size of the area under the curve $\bar{S}_i$ is calculated as:

$$area_i = \sum_{j=1}^{M} \bar{S}_{i,j} \tag{4.7}$$

where M denotes the top M nearest neighbors in each search score. We introduce the parameter M,to prevent the situation where the sorted curve from a good feature may go under that from a bad search score for a large M (as shown in Figure 4.3, the marked yellow line). This parameter controls the size of the area, and it is set as 400 in the experiments. Clearly, the calculated size of the area under each normalized score curve can be used to select the top N high quality features. We further assign an adaptive weight value to each of the top N scores:

$$weight_i = \frac{1}{area_i} \tag{4.8}$$

Finally, the fused search score from high quality deep binary codes is calculated as follows:

$$Score = \sum_{i=1}^{N} (S_i \times weight_i) \tag{4.9}$$

The proposed dynamic (top N) score-level late fusion scheme is adaptive and the quality of the deep binary code is automatically measured in an unsupervised manner. Clearly, it does not need any offline computation, thus the late fusion scheme is compatible with dynamic databases and suitable for large scale image search.

4.4 Experiments and Setup

In this section, we construct experiments and present the performance of our proposed image representation based on deep binary codes as well as the dynamic late fusion scheme in image retrieval.

The VGG network (VGGNet) [51] is employed to generate the deep binary codes (without fine-tuning). The deep convolutional layers: pool3, pool4, conv5 and pool5 from the VGGNet architecture are examined and the activations extraction from each deep convolutional layer is implemented using Caffe [153]. All images are resized to 224×224 before passing through the CNN network. The dimensions and the number of feature maps of the examined deep layers are summarized in Table 4.1. Max-pooling, sum-pooling, multi-scale-max-pooling, and multi-scale-sum-pooling are utilized to transform the activations from the feature maps to deep convolutional features, which are referred to as MP, SP, MSMP and MSSP, respectively. The deep binary codes are accordingly referred to as BMP, BSP, BMSMP and BMSSP. The cosine similarity measure is used to compare two images represented by their deep convolutional features (floating point values), while the Hamming distance is employed to compare the similarity based on the proposed deep binary codes (a binary string).

The experimental environment for the evaluation is a computer with an i7 CPU, 64GB of RAM, and an NVIDIA K40.

The source code of our bit-scalable deep binary codes and dynamic late fusion are released online at: http://press.liacs.nl/researchdownloads/.

Convolutional layer	The size of feature map	The number of feature maps
pool3	28×28	256
pool4	14×14	512
conv5	14×14	512
pool5	7×7	512

Table 4.1: Overview of the deep convolutional layers.

4.4.1 Datasets

We evaluate the performance of the deep binary code and the dynamic late fusion scheme on four publicly available datasets: INRIA Holidays [154], Oxford5K [122], UKbench [40] and MIRFLICKR 1M [136].

INRIA Holidays: this dataset consists of 1491 personal holiday photos that can be divided into 500 image groups, where the first image of each group is the query. The retrieval performance is measured in terms of mean Average Precision (mAP).

Oxford5K: this is a dataset composed of 5062 images which are downloaded from Flickr by searching 11 buildings or landmarks associated with Oxford. There are a total of 55 queries corresponding to 11 buildings and the performance is measured using mAP over the queries.

UKbench: a total of 10200 images are contained in this dataset, divided into 2550 groups. Each image is taken as the query in turn. The performance is measured by the average recall of the top four ranked images, referred to as N-S score.

MIRFLICKR 1M: this dataset includes one million images which are randomly retrieved from Flickr. We use this dataset to test the scalability of our deep binary code.

4.4.2 Evaluation of Deep Convolutional Feature Representation

We first evaluate the performance using deep convolutional representations (MP, SP, MSMP and MSSP), where the feature vectors generated by the operations of MP, SP, MSMP and MSSP are l_2-normalized. Table 4.2 summarizes the performance of the deep convolutional representations on each examined layer. For the single-scale pooling process, we observe that the sum-pooling operation achieves a better performance than max-pooling, while the multi-scale pooling scheme outperforms the single-scale operation. In general, the scheme of MSMP obtained the best search scores on each of the four benchmark datasets. It is also worth noting that the search precision from the lower layers to higher layers reveals an increasing trend, and the deep convolutional features on pool5 outperform features taken from other layers. This is mainly because each activation from higher deep layers correspond to a larger local region in the original image than those

from a lower deep layer, hence more semantic information is represented at a high deep layer.

Dataset	Deep feature	pool3	pool4	conv5	pool5
Holiday dataset (mAP score)	MP	57.24	71.40	78.50	79.27
	SP	65.23	75.49	76.37	79.17
	MSMP	67.22	76.91	80.24	**80.65**
	MSSP	64.40	74.80	76.18	78.81
Oxford5K dataset (mAP score)	MP	22.67	31.70	46.38	49.03
	SP	31.68	47.28	54.73	56.55
	MSMP	33.74	49.39	57.39	**58.05**
	MSSP	32.57	50.29	54.48	57.18
UKbench dataset (N-S score)	MP	2.7	3.34	3.72	3.73
	SP	2.95	3.47	3.7	3.73
	MSMP	3.04	3.54	3.74	**3.75**
	MSSP	2.94	3.46	3.66	3.7

Table 4.2: The performance of various deep convolutional features on image retrieval on the benchmark datasets. The accuracy is measured by the mAP score for the Holiday and Oxford5K datasets and the N-S score for the UKbench dataset.

4.4.3 Performance of Deep Binary Codes

We further test the image retrieval accuracy of the proposed deep binary codes on the benchmark datasets and the results are displayed in Table 4.3. The results demonstrate the effectiveness of the deep binary codes. We can see that the deep binary codes from the same layer generated using SP, MSMP and MSSP have similar performance on each dataset, and they all give better results than the representation based on MP. Compared to the performance of the deep convolutional features in Table 4.2, the dimensions of deep binary code are significantly reduced from 256 float values (2048 bytes of memory) to 256 bits (32 bytes of memory) on pool3 layer and 512 float values (4096 bytes) to 512 bits (64 bytes) on pool4, conv5, and pool5 layers, respectively. Meanwhile, the computation-time cost of the cosine similarity between two deep convolutional features (512

float values) is 0.14ms, while the comparison of the Hamming distance measure between two deep binary codes (512 bits) costs 0.007ms computation-time. The performance of deep binary codes is very competitive to deep convolutional features on the Holiday and UKbench datasets, which verifies that the deep binary codes have significant advantages with respect to speed/storage trade-off over the deep convolutional features, especially in the case of large scale image search.

Dataset	Deep feature	pool3	pool4	conv5	pool5
Holiday dataset (mAP score)	BMP	45.02	63.32	70.98	71.05
	BSP	59.47	73.04	74.52	**75.5**
	BMSMP	60.47	73.79	74.83	74.69
	BMSSP	57.78	73.82	72.94	74.65
Oxford5K dataset (mAP score)	BMP	21.4	33.93	43.13	42.59
	BSP	29.78	46.54	49.26	49.92
	BMSMP	29.1	46.68	48.93	49.55
	BMSSP	29.3	47.26	50.33	**50.45**
UKbench dataset (N-S score)	BMP	2.26	3.1	3.54	3.56
	BSP	2.83	3.41	3.62	3.64
	BMSMP	2.85	3.45	3.63	**3.64**
	BMSSP	2.84	3.42	3.59	3.62

Table 4.3: The performance of various deep binary codes on image retrieval based on four benchmark datasets. The accuracy is measured by mAP score for the Holiday and Oxford5K datasets and N-S score for the UKbench dataset.

4.4.4 Comparison with Hashing Learning Approaches

In the research literature, the closest related competitive algorithms are the unsupervised hashing learning methods [20, 23, 137, 138, 139, 140]. Specifically, to make a trade-off towards accuracy, efficiency and storage requirements in large scale image retrieval, hash function learning methods map deep convolutional features to binary string representations. In this section, we evaluate the performance of bit-scalable deep binary codes by comparing them with seven unsu-

Method	Holiday dataset (mAP)											
	pool3 (bits)			pool4 (bits)			conv5 (bits)			pool5 (bits)		
	256	128	64	512	256	128	512	256	128	512	256	128
BMSMP	**59.86**	**54.17**	**42.33**	**73.91**	**71.42**	**64.32**	**74.79**	**70.77**	**65**	**75.32**	**72.22**	**64.67**
LSH	44.48	34.13	23.64	69.42	59.93	53.33	71.57	66.75	58.49	69.47	67.69	57.82
SKLSH	37.2	25.74	23.37	64.62	53.93	39.32	67.78	60.7	46.56	66.97	60.53	47.17
ITQ	39.73	27.63	15.7	71.99	64.2	52.07	74.26	70.39	63.72	74.5	72	64.15
PCAH	12.05	18.11	26.4	37.98	48.3	48.17	46.56	62.48	62.96	44.14	60.49	62.65
SH	57.9	52.81	41.39	71.52	69.61	63.96	72.48	70.1	64.75	71.29	72.05	64.44
PCA-RR	43.98	28.78	30.82	68.29	64	53.13	73.17	69.69	64.02	72.71	70.66	64.05
DSH	53.02	46.24	37.84	63.9	60.7	53.5	66.32	60.84	55.5	67.99	63.86	57.31

Table 4.4: Comparison with various unsupervised hash function learning methods on the Holiday dataset.

Method	Oxford dataset (mAP)											
	pool3 (bits)			pool4 (bits)			conv5 (bits)			pool5 (bits)		
	256	128	64	512	256	128	512	256	128	512	256	128
BMSMP	**29.1**	**22.6**	**19.06**	**46.68**	**40.71**	**35.41**	**48.93**	**48.02**	**41.45**	49.55	47.36	40.51
LSH	21.35	15.84	4.21	40.57	30.33	23.4	47.2	46.07	39.85	48.72	44.78	33.51
SKLSH	16.32	14.04	7.52	39.45	26.88	23.98	46.47	38.53	29.75	**50.15**	39.96	33.6
ITQ	12.44	7.17	6.55	42.36	34.67	21.98	48.42	47.99	41.29	48.8	47.3	40.99
PCAH	10.6	12.13	7.01	20.27	23.9	27.08	33.17	37.77	37.58	35.08	41.54	38.45
SH	24.31	22.25	17.01	44.73	40.21	34.84	48.64	47.31	41.44	49.17	**48.4**	**42.84**
PCA-RR	17.56	16.76	13.01	38.59	31.4	27.39	48.9	47.91	39.53	49.14	47.07	40.49
DSH	21.89	20.22	17.75	32.2	28.25	2.43	40.78	37.58	30.6	43.32	38.06	28.78

Table 4.5: Comparison with various unsupervised hash function learning methods on the Oxford5k dataset.

pervised hash function learning methods. The compared approaches include two categories: data-independent methods (LSH and SKLSH) and data-dependent methods (ITQ, PCAH, SH, PCA-RR and DSH). The implementation of these methods are provided by the authors. Considering that the image representation based on MSMP achieves the best performance (as the results demonstrated in Table 4.2) and in order to make an objective comparison, all evaluated hashing learning methods map the deep convolutional features generated by using the MSMP operation. Moreover, different sizes of binary representations are evaluated.

Table 4.4, 4.5 and 4.6 illustrate the search accuracy of all the evaluated approaches on the benchmark datasets with different numbers of bits. We observe that deep binary codes from deep convolutional layers pool3, pool4 and conv5 give better results than the other hashing learning methods. The deep binary codes with different bit sizes from all examined layers obtained the best results on both the Holiday and UKbench datasets. The deep binary codes with 512, 256 and 128

Method	UKbench dataset (N-S score)											
	pool3 (bits)			pool4 (bits)			conv5 (bits)			pool5 (bits)		
	256	128	64	512	256	128	512	256	128	512	256	128
BMSMP	**2.6**	**2.6**	**2.18**	**3.46**	**3.31**	**2.66**	**3.64**	**3.47**	**3.2**	**3.65**	**3.48**	**3.21**
LSH	2.39	1.76	1.16	3.31	3.0	2.41	3.50	3.32	2.97	3.52	3.32	2.94
SKLSH	2.18	1.74	0.84	3.15	2.76	2.18	3.39	3.10	2.51	3.38	3.04	2.58
ITQ	1.78	0.89	0.33	3.22	2.82	2.19	3.54	3.4	3.11	3.55	3.41	3.12
PCAH	1.59	1.61	1.48	2.77	2.64	2.35	3.42	3.38	3.19	3.41	3.38	3.18
SH	2.6	2.58	2.12	3.43	3.21	2.65	3.58	3.45	3.11	3.59	3.45	3.18
PCA-RR	2.40	2.06	1.45	3.31	3.08	2.63	3.54	3.43	3.19	3.56	3.42	3.15
DSH	2.42	2.18	1.92	3.03	2.84	2.61	3.41	3.21	2.91	3.41	3.25	2.99

Table 4.6: Comparison with various unsupervised hash function learning methods on the UKbench dataset.

bits on pool5 show competitive performance to SKLSH and SH on the Oxford5K dataset. Regarding the data-dependent hash function learning approaches, the computational complexity and the time-cost will be significantly increased when the amount of training data becomes large. The deep binary code does not suffer from this issue because it does not need retraining. Furthermore, it shows its competitiveness and in some cases even better performance on image search compared to the hash function learning approaches.

4.4.5 Evaluation of the Late Fusion Scheme

In this section, we verify the effectiveness of the proposed dynamic top N score-level late fusion approach. Both binary string features and float value features are evaluated.

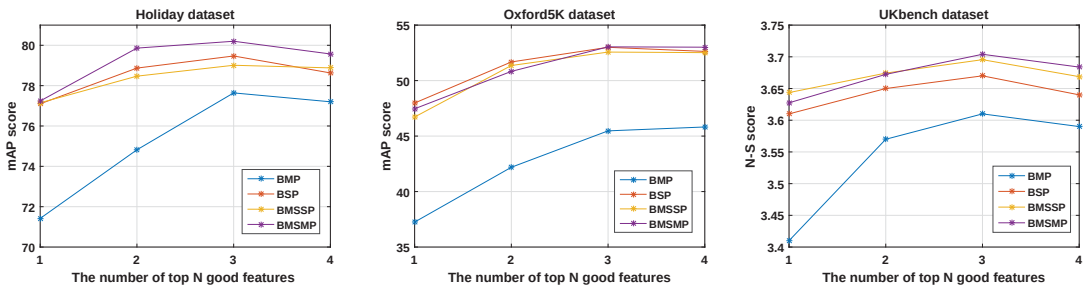


Figure 4.4: The search accuracy for different values of N. Four search scores from each of the compared methods are used, and most of them obtain the best fused score at value 3.

Method	Holiday dataset (mAP score)		Oxford5K dataset (mAP score)		UKbench dataset (N-S score)	
	Best	Fused	Best	Fused	Best	Fused
BMSMP	**75.32**	**80.3**	49.55	53.04	**3.65**	**3.71**
LSH	71.57	77.17	48.72	51.53	3.52	3.63
SKLSH	67.78	75.27	**50.15**	**54.27**	3.39	3.59
ITQ	74.5	77.85	48.8	50.75	3.55	3.61
PCAH	46.56	55.54	35.08	39.15	3.42	3.59
SH	72.48	79.52	49.17	54.17	3.59	3.68
PCA-RR	73.17	77.84	49.14	52.43	3.56	3.65
DSH	67.99	72.52	43.32	45.12	3.41	3.45

Table 4.7: The comparison of each evaluated method on image retrieval accuracy with and without top N score-level late fusion ($N = 3$).

The impact of the parameter N. First, we construct experiments to validate the influence of parameter N introduced in Formula (4.9). The normalized and sorted search scores from the deep binary codes generated by the operations of sum-pooling, max-pooling, multi-scale-sum-pooling and multi-scale-max-pooling on deep convolutional layers pool3, pool4, conv5 and pool5 are used. The dynamic late fusion for the deep binary codes is based on Formula (4.9) and the search accuracy on each test dataset is depicted in Figure 4.4. We find that the fusion accuracy from each search score increases steadily with N, while slightly decreasing at position *top*4. All the fused search scores show peak values at position *top*3, therefore, we set N equal to $\lambda - 1$, where λ is the number of fused features.

Then, we compare the retrieval performance of the binary string representation with the dynamic late fusion framework to the retrieval performance of the binary string representation without the dynamic late fusion framework. The deep binary codes and learned binary codes by hash functions are evaluated, and 256 bits on pool3, 512 bits on pool4, conv5 and pool5 are used in this comparison. The comparison results are shown in Table 4.7, "Best" denotes the best accuracy of each method from deep convolutional layers, "Fused" denotes that the search score is obtained using the dynamic late fusion scheme. We find that the search

accuracy is significantly increased after the *top*3 late fusion. The deep binary codes obtain the best fused scores on the Holiday and UKbench datasets and show competitive results on the Oxford5K dataset. Moreover, the fused scores also show competitive performance when compared to deep convolutional features.

Comparison with other fusion schemes. In order to further verify the strength of our late fusion method, we evaluate the dynamic late fusion scheme on some search scores using real valued features and compare the retrieval accuracy with two state-of-the-art late fusion schemes: graph model late fusion [152] and query-adaptive late fusion [148]. The comparison is carried out on the Holiday and UKbench datasets, and using the features of BoW (a 20K visual words histogram generated from rootSIFT [104] local descriptors and the *tf-idf* weight scheme), GIST [155] (a 512-dimensional global GIST descriptor), CNN [153] (a 4096-dimensional feature extracted from the first fully connected layer in the Alex CNN architecture), RAND (a global descriptor generated through multiplying by a random transform matrix) and HS (a 1000-dimensional HSV color histogram), respectively. The implementation of search scores on the Holiday and UKbench datasets from the five category features are offered by [156].

Formula (4.9) fuses the sorted search scores from binary string features using the Hamming distance to measure the similarity. We then modified it in Formula (4.10) to satisfy the distribution of search scores from float value features (BoW, GIST, CNN, HS and RAND) when using the cosine distance to measure the similarity. The N in Formula (4.10) is set to 4, because the number of search scores is 5 and we should set it equal to $5 - 1 = 4$.

$$Score = \prod_{i=1}^{N} (S_i)^{weight_i} \tag{4.10}$$

For graph model late fusion and query-adaptive late fusion, we use the code released from the papers [152] and [148] respectively. In order to make an objective comparison, the parameter M in Formula (4.7) is set to 400 such that it is equal to the corresponding parameter in the query-adaptive late fusion scheme. On the Holidays dataset, our late fusion scheme outperforms graph model fusion and

Method	Holiday dataset (mAP score)	UKbench dataset (N-S score)
Graph model [152]	81.04	3.82
Query-adaptive [148]	87.98	3.84
Ours	**88.61**	**3.84**

Table 4.8: Results on benchmarks with different fusion methods. We compare our method with Graph Fusion [152] and Query-adaptive [148] approaches.

the query-adaptive method. On the UKbench dataset, our result is equal to the query-adaptive method and better than graph model fusion. The comparison results further illustrate that the proposed dynamic top N late fusion method is effective for the search scores from both the binary string representation and the real valued representation.

4.4.6 Performance on Large Scale Image Search

In order to evaluate the performance of deep binary codes on large scale image search, we further perform large-scale experiments by combining the MIR-FLICKR 1M dataset with the Holiday, Oxford5K, and UKbench datasets. The deep binary codes as well as the binary codes learned by hash functions with a bit size of 256 on pool3, and a bit size of 512 on pool4, conv5 and pool5 are utilized for the evaluation. The accuracy results and the average time-cost of learning the hash function are summarised in Table 4.9. On each of the datasets with more than one million images, the deep binary code obtains the best accuracy with and without the dynamic late fusion scheme. This is further showing that the deep binary code is suitable for large scale image search and the dynamic late fusion scheme could significantly improve the search accuracy without requiring offline calculation.

Method	Holiday+1M (mAP)		Oxford5K+1M (mAP)		UKbench+1M (mAP)		Learning time cost
	Best	Fused	Best	Fused	Best	Fused	average(s)
BMSMP	**71.76**	**77.19**	**49.18**	**52.04**	**90.48**	**91.56**	-
LSH	60.68	69.26	44.24	47.3	83.26	87.36	-
SKLSH	55.33	66.99	46.1	51.18	79.79	87.19	-
ITQ	57.47	64.32	43.31	44.75	83.02	86.23	1120
PCAH	64.71	72.45	46.7	49.95	83.92	89.01	125
SH	64.3	73.98	46.08	50.38	86.4	90.28	1500
PCA-RR	61.91	70.27	45.44	49.38	85.44	88.77	190
DSH	52.13	59.13	38.8	39.8	76.98	78.69	400

Table 4.9: Comparison of the accuracy of each evaluated method for large scale image retrieval with and without top N score-level late fusion ($N = 3$), score-level late fusion (N = 3), and the time-cost of learning the hashing function.

Method	#dimensions	Holiday dataset (mAP score)	Oxford5K dataset (mAP score)	Ukbench dataset (mAP score/N-S score)
VLAD+RootSift [4]	128float	62.5	44.8	–/–
VLAD+CSurf [157]	128float	73.8	29.3	83.0/–
mVLAD+Surf [157]	128float	71.8	38.7	87.5/–
FV+T-embedding [158]	128float	61.7	43.3	85.0/–
FV+T-embedding [158]	256float	65.7	47.2	86.3/–
Sum pooling+PCAW [65]	256float	80.2	**58.9**	–/3.65
Max pooling+l_1 dist [64]	256float	71.6	53.53	84.2/–
Deep fully connected [63]	256float	74.9	43.5	–/3.42
Deep fully connected+finetune [63]	256float	78.9	55.7	–/3.56
BMSMP	512bit	74.83	49.55	90.78/3.65
FBMSMP	1792bit	**80.3**	53.1	**92.15/3.71**

Table 4.10: Comparison with state-of-the-art compact image representations on three benchmark datasets. FBMSMP denotes deep binary codes after applying dynamic top N late fusion.

Method	#Dim	Memory cost (Flicker 1M)	Holiday+Flicker 1M (mAP score)	Oxford5K+Flicker 1M (mAP score)	Ukbench+Flicker 1M (mAP score)
VLAD+RootSift[4]	128float	0.48G	37.8	–	–
Geometric+VLAD[6]	128float	0.48G	60.7	43.8	–
BMSMP	512bit	0.06G	71.76	49.18	90.48
FBMSMP	1792bit	0.24G	**77.19**	**52.04**	**91.56**

Table 4.11: Comparison with state-of-the-art compact image representations on large scale dataset. FBMSMP denotes deep binary codes after applying dynamic top N late fusion.

4.4.7 Comparison with state-of-the-art

We then compare the image retrieval results from deep binary codes with some other important state-of-the-art low dimensional image features. The results are from the papers [64] and [65], and the comparison is displayed in Table 4.10 and Table 4.11. Note that, the size of deep binary codes is 256-bit on pool3, 512-bit on pool4, conv5 and pool5. The results show that our deep binary codes outperform hand-crafted image representations, such as VLAD and Fisher Vector, and even outperform some recent CNN-based features. Moreover, after applying the top N late fusion scheme on the deep binary codes, the performance has been further improved.

4.5 Conclusions

In this chapter, we proposed a novel image representation called deep binary codes which have several important advantages over deep convolutional feature representations, as they can be calculated using a generic transferred model and therefore do not require additional training unlike many of the competitive algorithms from the research literature. The experimental results on well-known datasets as well as a large scale dataset show that deep binary codes are competitive to state-of-the-art approaches and can significantly reduce memory requirements and computational costs for large scale image search. Second, the dynamic late fusion scheme estimates the quality of each feature in a query-adaptive manner which highlights the strengths of score-level fusion without needing supervision and offline calculations. In our experiments the dynamic late fusion scheme gave consistent improvements in accuracy.

Chapter 5

Comparison of Information Loss Architectures in CNNs

Recent advances in image classification have been achieved by the application of deep Convolutional Neural Networks (CNNs). Pooling and sub-sampling operations in the CNNs lead to invariance to local transformations, but result in loss of accuracy. In this chapter, we propose a novel deep neural network called the "Weighted Integration Architecture Network" (WIAN) that can effectively recover the information loss due to the pooling operations in the CNNs. The proposed WIAN reuses the information from the previous layers in the network and assigns a weight matrix to each layer; and then integrates them to further enhance the image classification performance. Two weight value generation schemes are investigated: the first one is calculated according to the responses or entropy in the layer, and the second one is an adaptive learning scheme. Exhaustive experiments on four standard benchmark datasets (CIFAR-10, CIFAR-100, MNIST and SVHN) demonstrate the effectiveness and improved performance of the proposed WIAN.

5.1 Introduction

Prior to convolutional neural networks, the commonly and widely used approaches in image classification were using the Bags-of-Words (BoW) model [159]. This type of model first encodes the local features from the salient regions in the image as a histogram of quantized visual words, and then feeds the histogram into a SVM classifier [160]. This method is a type of orderless statistics that incorporates spatial geometry into the BoW representation. Lazebnik et al. [77] integrated a spatial pyramid framework into the BoW feature generation, that counts the number of visual words inside a set of image sub-regions instead of the whole image region. This procedure was further improved by using sparse coding optimization for the construction of a visual vocabulary [161], obtaining the best performance on the ImageNet 1000-class classification problem. The approaches based on the visual word model can be viewed as zero order statistics (i.e., counts of visual words), and discard a lot of valuable information of the image. The Fisher Vector image representation introduced by Perronnin et al. [162] overcame this issue and extracted first and second order statistics by employing the Fisher Kernel [163], achieving state-of-the-art image classification results.

Recently, a significant performance gain on the task of image classification has been made with deep convolutional neural networks (CNNs) [164, 165]. This is mainly due to their ability to learn rich high level image representations as opposed to hand-designed low-level features, as well as the availability of very large and more comprehensive training data.

Traditional convolutional neural networks used for image classification consist of several stacked convolutional layers (optionally followed by a normalization layer and a pooling layer), fully connected layers and a softmax layer (a classifier) on the top. Convolutional layers take the inner product of a linear filter and the underlying receptive field followed by a nonlinear activation function at every local region of the input. The outputs from each convolutional layer are called feature maps. The fully connected layer has connections to all individual activations in the feature maps from the previous layer and the resulting vector can be fed into the softmax layer for classification (as shown in Figure 5.1). Variants

of this basic design are proposed to improve the performance of the network. Most recent methods increase the depth of the CNN architecture as well as the width of each layer to enhance the performance [47, 166]. However, increasing the depth of CNNs brings the issue of vanishing gradients and over-fitting during the optimization of the network, especially if the number of labeled examples in the training set is limited. Several useful technologies are employed to address the issue of over-fitting: data augmentation which increases the number of training samples when using a small dataset, pre-training which initializes the networks with pre-trained parameters rather than randomly set parameters, and dropout which randomly omits half of the feature detectors, aims to prevent complex co-adaptations on the training data and enhance the generalization ability. The architecture of GoogleNet [42] is designed such that the depth and width of the network is increased while the computational budget is keep constant. The Network-in-Network (NIN) is an approach proposed by Lin et al. [47] that replaces the linear convolution by a nonlinear convolution function to enhance the abstraction ability of the neural network. Deeply supervised networks [167] focus on the importance of minimizing the output classification error while reducing the prediction error of each individual layer. A Siamese network [168] is trained with a pairwise loss function that minimizes the distance between the same class and maximizes the distance between different classes. A similar triplet network [169] employs the triplet ranking loss function to preserve relative similarity relations.

In this chapter, we propose a novel architecture called Weighted Integration Architecture Network (WIAN) to boost the performance of image classification. WIAN starts by reshaping the convolutional layers to the same shape by a convolution operation, and normalizes each reshaped convolutional layers to the same scale. WIAN automatically learns a weight value matrix using an adaptive learning scheme or using the responses or entropy on each reshaped and normalized convolutional layer. Then these convolutional layers are multiplied by the assigned weight matrixes respectively, and finally combined into a single layer by element-wise summing, as illustrated in Figure 5.3. The main contributions of WIAN are as follows:

First, the weight matrix learning scheme for each layer is adaptive.

Second, the integration layer can effectively recover the spatial information loss caused by the pooling operation and improve the accuracy of image classification.

The remainder of this chapter is organized as follows: we make a review of related work on the recovery of spatial information loss in Section 5.2. Section 5.3 gives an overview of convolutional neural networks for image classification. Section 5.4 provides a detailed description of the proposed Weighted Integration Architecture Network (WIAN). Section 5.5 presents the experimental results, and conclusions are given in Section 5.6.

5.2 Related Work

In CNNs, a convolutional layer is usually followed by a pooling operation. The pooling operation reduces the spatial resolution by computing a summary statistic over a local spatial region (typically a max or average operation). The main motivation behind the use of pooling is to promote invariance to local input transformations (such as translation, occlusion and truncation of the local stimulus). This is mainly due to the fact that the resulting outputs after pooling show invariance to the spatial location within the pooling region. Hence, the pooling layer is particularly important for the performance of image classification where local image transformations may obfuscate the object identity. Additionally, the pooling layer plays a vital role in preventing over-training while reducing computational complexity for the task of image classification. However, these invariance achieved by pooling come at the price of loss of accurate spatial information. Several research efforts attempt to make up for the loss caused by the pooling operation. A commonly used method is cascaded convolutional neural network. Sun et al. [170] proposed to use cascaded convolutional networks to improve the accuracy of facial landmarks detection and Toshev et al. [71] applied the cascaded convolutional network to the human pose estimation. Tompson et al. [75] designed a heat-map regression model to refine the locations of human body joints. Yang

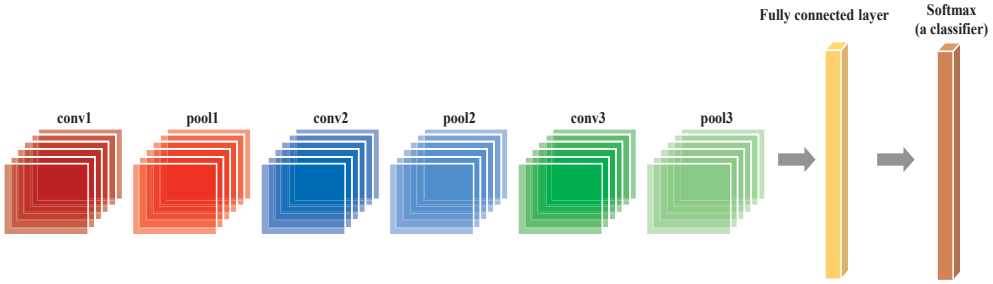


Figure 5.1: The architecture of a standard deep Convolutional Neural Network (CNN) used for image classification.

et al. [171] designed a DAG-CNN which extracts multi-scale features across each layer in the CNN and further integrates them for image classification. Inspired by the architecture of DAG-CNNs, the proposed WIAN automatically learns a weight matrix for each of previous layers and then integrates them for image classification. Thereby, the WIAN could improve the performance of the CNN.

5.3 Convolutional Neural Networks Classification

Considering a standard CNN architecture, as depicted in Figure 5.1, there are N convolutional layers, denoted as C^1,..., C^N. Each convolutional layer is followed by a pooling layer denoted as P^1,..., P^N, respectively. The objective of training a traditional CNN is to maximize the probability of the correct class, which is achieved by minimizing the softmax loss function. For a specific training set which includes m images: $\{(I^{(i)}, L^{(i)}); i = 1, ..., m\}$, where $I^{(i)}$ is the i^{th} image and $L^{(i)} \in \{1, ..., K\}$ is the class label. Let $\{x_j^{(i)}; j = 1, ..., K\}$ be the output of the activation j in the last fully connected layer, then the probability that the label of $I^{(i)}$ is j is given by

$$p_j^{(i)} = \frac{exp(x_j^{(i)})}{\sum_{j=1}^{K} exp(x_j^{(i)})} \tag{5.1}$$

The output of the fully connected layer is then fed into the softmax layer which aims to minimize the following loss function:

$$J_\theta = -\frac{1}{n}[\sum_{i=1}^{n}\sum_{j=1}^{K} 1\{L^{(i)} = j\} \log(p_j^{(i)})] \tag{5.2}$$

where $1\{.\}$ is the indicator function. Standard back-propagation is utilized to optimize the parameters of the network by computing the derivatives of the defined loss function.

Additionally, the success of AlexNet [48] suggests that the features emerging at the fully connected layers of a CNN trained for image classification can serve as good descriptors, when for example, using a SVM classifier for image classification.

5.4 Integration Architecture Network

As the architecture of standard CNNs did not take into account the information loss caused by the pooling operation, in this section, we explore several useful practices to integrate the information from the previous convolutional layers to recover the accuracy loss in CNNs. Performance evaluation results demonstrate that the integration of information from the previous convolutional layers could effectively increase the performance of image classification.

5.4.1 Concatenate Architecture Network

Inspired by the architecture of GoogleNet, a simple and effective way to train a high quality CNN is to concatenate the previous convolutional layer into a new layer. The illustration of the concatenate architecture network (CONCAT) is shown in Figure 5.2. In this architecture, we first reshape the convolutional layers in the CNN into the same shape by applying a convolution operation. These reshaped layers are normalized into the same scale and concatenated together. The fully connected layer takes all outputs of neurons in the concatenated layers

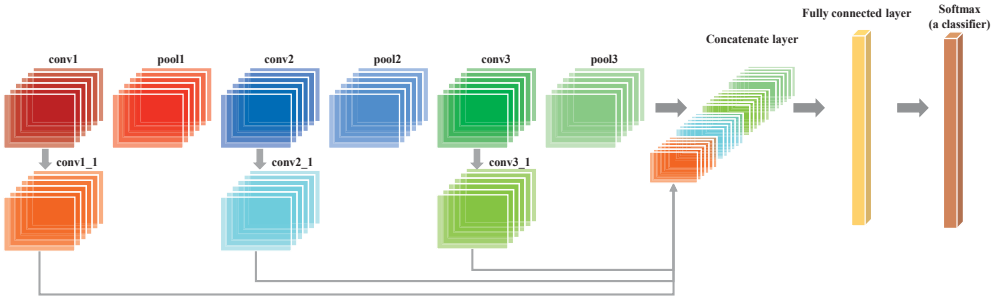


Figure 5.2: The illustration of the concatenate network (CONCAT). Each convolutional layer is reshaped and normalized, and concatenated in one layer for further processing.

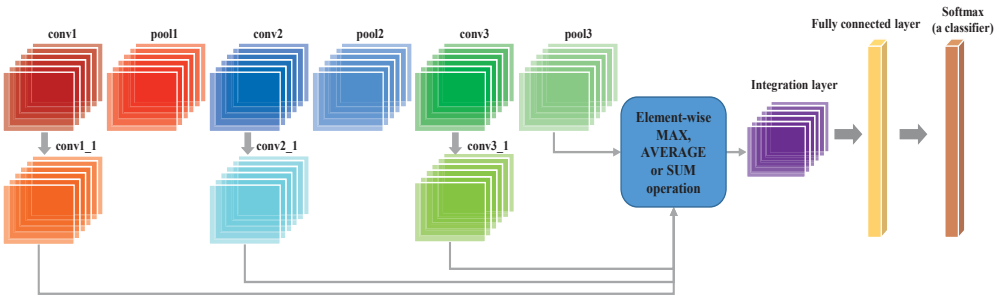


Figure 5.3: The architecture of the proposed Integration Architecture Networks. The layers are integrated by element-wise max, average or sum operations, respectively. The resulting network is called Max Integration Architecture Network (MIAN), Average Integration Architecture Network (AIAN) and Sum Integration Architecture Network (SIAN), respectively.

as input to every single neuron it has. Finally the output from the fully connected layer is fed into the softmax loss function optimizing classification.

5.4.2 Weighted Integration Architecture Network

The concatenate operation significantly increases the width of the integration layer, which means a larger number of parameters are stored in this layer. However, a large amount of parameters results in high storage requirements, and also it makes the network susceptible to over-fitting, especially if the amount of

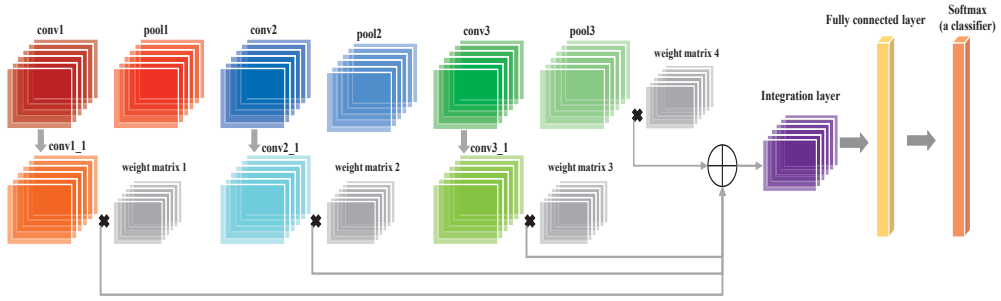


Figure 5.4: The architecture of the proposed Weighted Integration Architecture Network (WIAN). A weight matrix is assigned to each integration layer.

labeled data in the training set is small. Additionally, because of the existing redundant information between two adjacent layers, we propose to integrate the previous convolutional layers (reshaped to the same dimensions by a convolution operation and subsequently normalized) by applying element-wise max, average or sum operations, as depicted in Figure 5.3. The resulting network architectures are named the Max Integration Architecture Network (MIAN), the Average Integration Architecture Network (AIAN) and the Sum Integration Architecture Network (SIAN), respectively. Furthermore, we propose an adaptive method to integrate the previous convolutional layers, which assigns to each previous convolutional layer a weight matrix, respectively, and then combine them by element-wise summing (as shown in Figure 5.4). Two weight schemes are explored in this section, one relies on the responses or entropy of the convolutional layer and the other one is based on an adaptive weight learning method.

Responses based weight scheme: as shown in Figure 5.4, for the given N convolutional layers in the network, we denote the feature maps from layer C^n as $F^n, n = 1, ..., N$. These feature maps can be represented as a vector with dimension $w^n \times h^n \times c^n$, where w^n and h^n are the width and height of each individual feature map, and c^n denotes the number of feature maps of layer C^n. We further associate each unit of a feature map with a spatial coordinate (x, y) and the activation of this unit by $a(x, y)$. The response value of each feature map is calculated as $r^c = \sum_{x=1}^{w^n} \sum_{y=1}^{h^n} a(x, y), c = 1, ..., c^n$. The response value of each layer is computed as $R^n = \sum_{c=1}^{c^n} r^c$. The weight value in the weight matrix of

each layer is defined as:

$$weight_R^n = \frac{R^n}{\sum_{n=1}^{N} R^n} \tag{5.3}$$

Entropy based weight scheme: we further employ entropy information [172] on each convolutional layer to define a weight value. The activation of each unit $a(x, y)$ in the feature map can be treated as a state p_i, and the entropy of each feature map is computed by $e^c = \sum_{x=1}^{w^n} \sum_{y=1}^{h^n} (p_i \times \log p_i), c = 1, ..., c^n$. The entropy of each layer is computed as $E^n = \sum_{c=1}^{c^k} e^c$. The weight value in the weight matrix of each layer is then defined as:

$$weight_E^n = \frac{E^n}{\sum_{n=1}^{N} E^n} \tag{5.4}$$

For the responses or entropy based weight scheme, each unit in the weight matrix is assigned the same value of response or entropy calculated from each layer, thus, the weight matrix of responses or entropy based weight scheme can be reduced to one single weight value.

Finally, the activation value of each unit $a^{n+1}(x, y)$ in the integration layer is calculated using the following formula:

$$a^{n+1}(x, y) = \sum_{n=1}^{N} weight^n \times a^n(x, y) \tag{5.5}$$

Note that, in the specific case that the weight value from each layer is equal to $1/N$, the scheme becomes equal to the scheme of AIAN. If the weight from each layer is equal to 1, the scheme becomes equal to the scheme of SIAN.

Adaptive weight learning scheme: we further investigate an adaptive weight learning scheme in this chapter. Different from the responses or entropy based scheme where each unit in the weight matrix share the same weight value, the adaptive weight learning scheme assigns to each of the integrated layer a weight matrix. The initial values in each weight matrix are set to $1/k$, where k is the number of integrated layers. Then each unit in the weight matrix is automatically updated during each iteration of the CNN training. Let $weight_{(x,y)}^n$ be the value

of each unit in the weight matrix of the n^{th} layer, then the integrated value of each unit $a^{n+1}(x, y)$ in the integration layer is calculated as:

$$a^{n+1}(x,y) = \sum_{n=1}^{N} weight^{n}_{(x,y)} \times a^{n}(x,y) \tag{5.6}$$

5.5 Experimental Results

The proposed WIAN is implemented using Caffe [153]. The experimental environment is consisted of a computer with an i7 processor, 32GB RAM, and an NVIDIA TITANX. The network is trained using mini-batches of size 100 without data augmentation. The training process starts from the initial weights and learning rates, and it continues until the accuracy on the training set stops improving. Then the learning rates are lowered by a factor of 10 according to an epoch schedule determined on the validation set. The source code of WIAN is available at: http://press.liacs.nl/researchdownloads/.

5.5.1 Datasets

We evaluate the performance of WIAN on four benchmark datasets: CIFAR-10 [164], CIFAR-100 [164], MNIST [8] and SVHN [173].

CIFAR-10: the CIFAR-10 dataset is constructed for object recognition. It is composed of 10 object classes, with 6000 images per class. 50000 images are selected for training, and the remaining 10000 images are used for testing. Each image is given in the RGB-format with size 32×32 pixels.

CIFAR-100: the CIFAR-100 dataset is similar to the CIFAR-10 dataset (both use the same image size and format), except that the CIFAR-100 contains 100 classes with 600 images per class. CIFAR-100 also uses 50000 images for training and the remaining 10000 images for testing.

MNIST: the MNIST dataset consists of images of hand written digits which are 28×28 pixels in size. There are 60000 training images and 10000 testing images

in total. For this experiment, all images of the dataset have been resized to a fixed resolution of 32×32 pixels.

SVHN: the Street View House Numbers (SVHN) dataset is a collection of house numbers in the Google Street View images. It is composed of over 600000 color images with a fixed resolution of 32×32 pixels.

5.5.2 Details of Weighted Integration Architecture

The architecture of the network in the evaluation contains three convolutional layers, followed by Rectified Linear Unit (RELU) normalization and pooling operations, as well as a fully connected layer and a softmax classifier on top. Moreover, in order to integrate the previous convolution layers into one layer, we first convolute them to the same shape, normalize them into the same scale and then combine them.

According to the parameter configuration of each layer, the architecture of the WIAN in the performance evaluation can be described concisely by layer notations with the following layer sizes (CONV denotes the convolutional layer, RELU denotes the rectified linear unit layer, POOL denotes the pooling layer, and FC denotes the fully connected layer):

$INPUT(32 \times 32 \times 3)$

$CONV1(32 \times 32 \times 32) \rightarrow RELU1 \rightarrow POOL1(16 \times 16 \times 32)$

$CONV2(16 \times 16 \times 32) \rightarrow RELU2 \rightarrow POOL2(8 \times 8 \times 32)$

$CONV3(8 \times 8 \times 64) \rightarrow RELU3 \rightarrow POOL3(4 \times 4 \times 64)$

$CONV1 \rightarrow CONV1\_1(4 \times 4 \times 64)$

$CONV2 \rightarrow CONV2\_1(4 \times 4 \times 64)$

$CONV3 \rightarrow CONV3\_1(4 \times 4 \times 64)$

$CONV1\_1 + CONV2\_1 + CONV3\_1 + POOL3 \rightarrow FC \rightarrow Softmax$

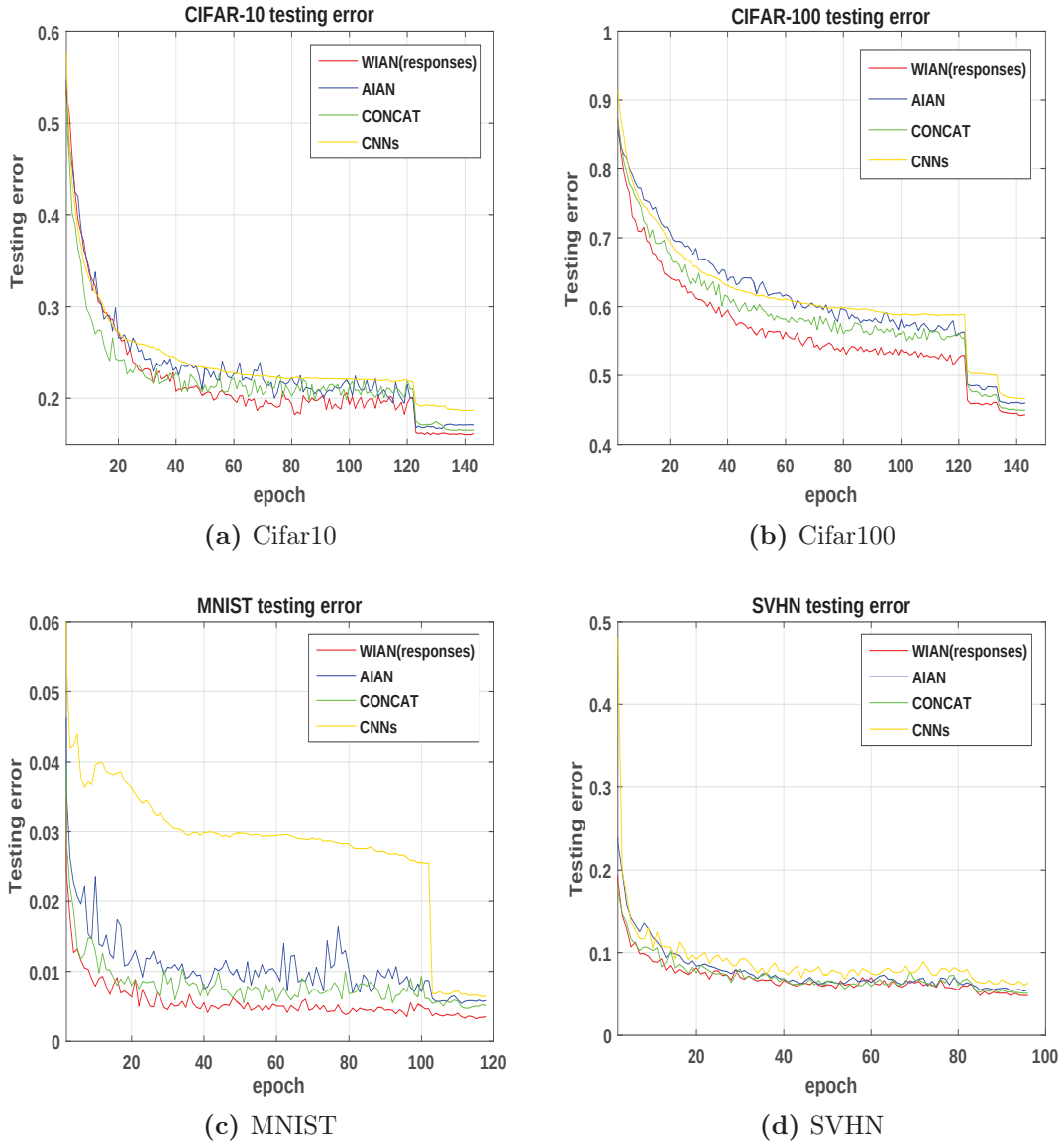
Methods	CIFAR-10	CIFAR-100	MNIST	SVHN
WIAN(responses)	**83.92**	**55.84**	**99.65**	**95.21**
WIAN(entropy)	83.86	55.25	99.58	94.95
WIAN(adaptive learning)	83.15	54.25	99.55	94.6
MIAN	82.75	54.2	99.4	94.55
AIAN	82.9	54.1	99.46	94.68
SIAN	82.6	53.9	99.3	94.52
CONCAT [42]	83.3	55.06	99.51	94.94
CNNs [48]	81.5	53.5	99.3	94.15

Table 5.1: The performance comparison of different convolutional neural network architectures on the four benchmark datasets, CIFAR-10, CIFAR-100, MNIST and SVHN. The number in the table denotes the accuracy of image classification.

5.5.3 Evaluation Results

We present the performance of our proposed WIAN (three weight schemes are evaluated, the first one is based on responses, the second one relies on entropy information and the third one is an adaptive weight learning scheme) and make a comprehensive comparison with general CNNs, Max Integration Architecture Networks (MIAN), Average Integration Architecture Networks (AIAN), Sum Integration Architecture Networks (SIAN) as well as the directly concatenate (CONCAT) of the previous convolutional layers in the CNN architecture. The concatenation operation is similar to the inception module in GoogleNet [42]. A softmax loss function is employed to predict the classification accuracy. The evaluation results of the classification accuracy are listed in Table 5.1.

It turns out that the evaluated integration schemes (WIAN, MIAN, AIAN, SIAN and CONCAT) all achieve improved performance when compared to general CNNs. The WIAN (based on responses, entropy and adaptive weight learning on each layer in the CNN) show much better results than the other approaches. WIAN based on the weight calculated according to the responses on each layer shows the best performance on all the benchmarks. The integration schemes of MIAN, AIAN and SIAN show similar results on the test datasets.



(a) Cifar10 **(b)** Cifar100

(c) MNIST **(d)** SVHN

Figure 5.5: The comparison of the classification error among several possible architectures on the four benchmark datasets.

Additionally, we further investigate the behaviours of the testing error during each epoch in the CNN training. The performance of WIAN (responses), AIAN, CONCAT and the general CNNs are evaluated. The graphs depicted in Figure 5.5 show that WIAN (responses) reaches the smallest testing error faster than others.

This further demonstrates that the weighted integration of previous convolutional layers can boost the performance of the network.

5.6 Conclusions

In this chapter, we propose to reuse the information encoded in previous layers in the network to recover the precision loss due to the pooling operation in the CNN. We present a novel Weighted Integration Architecture Network (WIAN) to enhance the performance of CNN based image classification, where each layer is multiplied by a weight matrix generated according to the responses or entropy of the layer, adaptive learning and then element-wise summed together. The evaluation results demonstrated that the WIAN can yield high accuracy on image classification, and WIAN shows better performance than the scheme that employs direct concatenation, and the schemes employing max, average and sum integration of the previous convolutional layers in the CNN architecture. Moreover, WIAN based on the weight value calculated according to the responses on each layer is more robust than WIAN based on entropy value as well as the adaptive learning scheme.

Chapter 6

Conclusions

6.1 Conclusions

In this thesis, we focus on large scale visual search. The topic of large scale visual search has seen a steady train of improvements in performance over the last decade. In this task, given a query image containing a specific object or scene, the goal is to return the images containing the same object or scene that may be captured from different viewpoints, under changed illumination and maybe occluded. The Bag-of-Words model was originally proposed for document retrieval. The introduction of salient point methods has made this model applicable to the image domain where it translates to the visual word model. General salient point methods involve a detector and a descriptor. The detector locates the salient regions in the image and the descriptor encodes discriminative information in the salient region into a local feature. Based on the salient point method, an image can be transformed into a collection of local feature vectors, which can be viewed as prototypes of words in text. The visual word model has been the state-of-the-art for many computer vision applications. It has greatly advanced the research of instance retrieval in the past ten years, and many improvements have been proposed.

One important aspect in the visual word model is the degree to which the salient point methods are invariant to image translation, scaling, and rotation, as well

as partially invariant to illumination changes, and robust to local geometric distortion. In Chapter 2, we presented a comparison of the existing salient point detectors and descriptors on diverse image distortions. These comparative experimental studies can benefit researchers in choosing an appropriate detector and descriptor for different computer vision applications. According to the evaluation results, we find that the FAST detector had the highest repeatability score compared to other detectors, moreover it had the lowest detection time-cost per point. Regarding the criterion of recall-precision, our experiments showed that the descriptors of SIFT, BRISK, and FREAK were the best performing affine invariant descriptors. Furthermore, evaluation of the time complexity showed that the binary descriptors are efficient with respect to feature description and matching.

Existing salient points methods tend to perform poorly to viewpoint changes. In Chapter 3, we presented the Retina-inspired Invariant Fast Feature, RIFF, which was designed for invariance to scale, rotation, and affine image deformations. The RIFF descriptor is based on pair-wise comparisons over a sampling pattern loosely based on the sampling pattern seen in the human retina and introduces a method for improving accuracy by maximizing the discriminatory power of the point set. The main contribution of the RIFF descriptor is in constructing the descriptor, where the discriminative power is optimized by ranking and deleting points with low distinctiveness. In our Bag-of-Words image retrieval tests on three well known datasets, RIFF outperformed the other feature descriptors with respect to invariance to scale, rotation, and affine transformations. Furthermore, we presented a performance evaluation of real valued and binary string salient point descriptors. The time complexity and space requirements showed that binary string descriptors are efficient in terms of feature extraction time and memory usage. With respect to the criterion of the mAP score, the image copy detection experiments showed some significant strength of binary string local descriptors: FREAK clearly outperformed SIFT on invariance to rotation, scale, and affine transformations; BRIEF had the best accuracy testing invariance to image blur and was among the best in robustness to cropping.

In recent years, the focus on image search has shifted from the visual word model to deep Convolutional Neural Networks (CNNs) features. The CNN is a hierarchical structure that has been shown to outperform hand-crafted features in a number of vision tasks, such as object detection, image segmentation, and classification. The power of CNNs mainly comes from the large number of parameters and the use of large scale datasets with rich annotations. Using the features extracted from CNN models, researchers have reported competitive performance compared to the classic visual word model. In Chapter 4, we proposed a novel image representation called deep binary codes which have important advantages over deep convolutional feature representations, as they can be calculated using a generic transferred model and therefore do not require additional training. The experimental results on well-known datasets as well as a large scale dataset show that deep binary codes are competitive to state-of-the-art approaches and can significantly reduce memory and computational costs for large scale image search. Moreover, in Chapter 5, we proposed to reuse the information in the previous layers in the network to recover the precision loss due to the pooling operation in the CNN. The presented Weighted Integration Architecture Network (WIAN) can enhance the power of the CNN model.

6.2 Future Work

In the future, we will try to improve our work in the following directions:

Convolutional neural networks based local descriptor generation: The generation of effective local image descriptors plays an important role in the applications of computer vision involving baseline stereo vision, structure from motion, visual words based image search, image classification and object detection, etc. The existing schemes of local descriptor generation can be categorized into hand-crafted or automatically learned schemes. Recent work focuses more on automatic learning of local descriptors. Learning based schemes usually optimize an objective function to generate robust and distinctive local descriptor. In particular, the most common objective functions are designed to minimize the

distance between the descriptors from the same 3D location (scale and location) or same class label extracted under varying imaging conditions and different viewpoints, and maximize the distance between patches from different 3D locations or different class labels. Concurrently, the automatically learning schemes of local descriptors based on deep convolutional neural networks have recently made dramatic progress. A Siamese network trained with a pair-wise loss ranking function and a triplet network trained with a triplet loss ranking function that also minimizes the distance (in the embedded space) between patches of the same labels and maximizes the distance between patches of different labels are used to automatically learn high performance local descriptors. However, all these methods suffer from huge training complexity, because they directly train CNNs using the pair-wise or triplet list, the length of which scales with the quadratic or cubic with the number of images in the training dataset. Therefore, it is important to further develop techniques to address huge training complexity while maintaining the robustness of the learned local descriptors. Another issue we need to address is the limitation of training data. The typical solution is to generate more training data from existing data using data augmentation schemes, such as scaling, rotating and cropping. Hence, it is important to further develop techniques for generating or collecting more comprehensive training data, which could make the networks learn better features that are robust to various changes, such as geometric transformations, and occlusion.

Convolutional neural networks based high level image representation: The outputs from the fully connected layer in the CNN are mostly used as image representation. However, the image representation from a fully connected layer suffers from the lack of description of local patterns, which is especially critical when occlusions or truncations exist in the images. With respect to the sensitivity to local stimulus, CNN features from the bottom or intermediate layers have shown promising performance. These discriminatively trained convolutional kernels respond to specific visual patterns that evolve from bottom to top layers. While capturing local activations, the intermediate features are less invariant to image translations. Compared to the pooling operation, which is usually utilized to map the intermediate features into global feature, one promising direction for

future research is to find more efficient ways to convert the intermediate features into low dimensional and high distinctiveness image representations, in order to avoid the information loss caused by pooling operations. Second, it is known that the top layers in CNNs are sensitive to semantics, while intermediate layers are specific to local patterns. For the image representation, we can obtain multiple layer features in the pre-trained CNN through one feed-forward step. It is not trivial to predict which layers are superior. Therefore, the fusion of the features from multiple layers is a good practice to further improve the accuracy of image search. Moreover, we can also fuse the features from different models to represent the image.

Convolutional neural networks based deep hash learning: In order to achieve efficient large scale image search, the high performance of the supervised deep hashing model appears to be promising. The first direction is to increase the ability to generalize by increasing the width or depth of the networks, for example, the width and depth of the CNN models in the literature [42, 51]. Larger networks could normally bring higher quality performance, but have the danger of over-fitting and require very large computational resources. A second direction is to define a good loss ranking function. As the commonly used pair-wise loss functions and triplet loss functions employ Euclidean distance to measure the similarity in the input space, we can replace the Euclidean distance with different similarity metrics for different input spaces. Moreover, we can also incorporate constraint information from the input space to the loss functions. A third direction towards more powerful models is to design more specific deep networks. Currently, almost all of the CNN-based schemes adopt a shared network for their predictions, which may not be distinctive enough. The study by Ouyang et al. [174] has verified that object-level annotation is superior to image-level annotation for object detection. This can be viewed as a kind of specific deep network that just focuses on the object region rather than the whole image. Another issue we need to note is that in some situations the amount of the annotated data is insufficient and it could result in over-fitting during the training of the CNN. Semi-supervised deep hashing makes use of the labeled data together with the

unlabeled data and may be able to overcome this limitation in the CNN training.

Bibliography

[1] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the 9th International Conference on Computer Vision. (2003) 1470–1477

[2] Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8

[3] 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

[4] Arandjelovic, R., Zisserman, A.: All about VLAD. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 1578–1585

[5] Delhumeau, J., Gosselin, P.H., Jégou, H., Pérez, P.: Revisiting the VLAD image representation. In: Proceedings of the 21st ACM international conference on Multimedia. (2013) 653–656

[6] Wang, Z., Di, W., Bhardwaj, A., Jagadeesh, V., Piramuthu, R.: Geometric VLAD for large scale image search. arXiv preprint arXiv:1403.3829 (2014)

[7] Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In: Proceedings of European Conference on Computer Vision. (2012) 774–787

[8] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86** (1998) 2278–2324

[9] 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 Conference on Computer Vision and Pattern Recognition. (2009) 248–255

[10] Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. Journal of Applied Statistics **21** (1994) 225–270

[11] Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th International Conference on Computer Vision. (1999) 1150–1157

[12] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Proceedings of European Conference on Computer Vision. (2006) 404–417

[13] Weickert, J., Romeny, B.T.H., Viergever, M.A.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Transactions on Image Processing **7** (1998) 398–410

[14] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

[15] Rosin, P.L.: Measuring corner properties. Computer Vision and Image Understanding **73** (1999) 291–307

[16] Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary robust invariant scalable keypoints. In: Proceedings of the International Conference on Computer Vision. (2011) 2548–2555

[17] Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast retina keypoint. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 510–517

[18] Wu, S., Lew, M.S.: RIFF: Retina-inspired invariant fast feature descriptor. In: Proceedings of the ACM International Conference on Multimedia. (2014) 1129–1132

[19] Chum, O., Matas, J.: Fast computation of min-hash signatures for image collections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 3077–3084

[20] Raginsky, M., Lazebnik, S.: Locality-sensitive binary codes from shift-invariant kernels. In: Advances in Neural Information Processing Systems. (2009) 1509–1517

[21] Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: International Conference on Very Large Databases. Volume 99. (1999) 518–529

[22] Chum, O., et al.: Large-scale discovery of spatially related images. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 371–377

[23] Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems. (2009) 1753–1760

[24] Shao, J., Wu, F., Ouyang, C., Zhang, X.: Sparse spectral hashing. Pattern Recognition Letters **33** (2012) 271–277

[25] Zhang, D., Wang, J., Cai, D., Lu, J.: Self-taught hashing for fast similarity search. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. (2010) 18–25

[26] Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: Proceedings of the 28th International Conference on Machine Learning. (2011) 1–8

[27] Irie, G., Li, Z., Wu, X.M., Chang, S.F.: Locally linear hashing for extracting non-linear manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 2115–2122

[28] Li, X., Lin, G., Shen, C., Van Den Hengel, A., Dick, A.R.: Learning hash functions using column generation. In: Proceedings of the International Conference on Machine Learning. (2013) 142–150

[29] Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 2074–2081

[30] Norouzi, M., Blei, D.M.: Minimal loss hashing for compact binary codes. In: Proceedings of the 28th International Conference on Machine Learning. (2011) 353–360

[31] Huang, L.K., Yang, Q., Zheng, W.S.: Online hashing. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. (2013) 1422–1428

[32] Norouzi, M., Fleet, D.J., Salakhutdinov, R.R.: Hamming distance metric learning. In: Advances in Neural Information Processing Systems. (2012) 1061–1069

[33] Wang, J., Kumar, S., Chang, S.F.: Sequential projection learning for hashing with compact codes. In: Proceedings of the 27th International Conference on Machine Learning. (2010) 1127–1134

[34] Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for large-scale search. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 2393–2406

[35] 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 International Conference on Computer Vision. (2015) 118–126

[36] Kumar, B., Carneiro, G., Reid, I.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. arXiv preprint arXiv:1512.09272 (2015)

[37] Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4353–4361

[38] Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3279–3286

[39] Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple object class detection with a generative model. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 1. (2006) 26–36

[40] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2006) 2161–2168

[41] Zeiler, M.D.: Hierarchical convolutional deep learning in computer vision. PhD thesis, NEW YORK UNIVERSITY (2013)

[42] 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 Conference on Computer Vision and Pattern Recognition. (2015) 1–9

[43] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 685–694

[44] Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th International Conference on Machine Learning. (2010) 111–118

[45] Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: International Conference on Artificial Neural Networks. (2010) 92–101

[46] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 580–587

[47] Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)

[48] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012) 1097–1105

[49] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

[50] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proceedings of European Conference on Computer Vision. (2014) 346–361

[51] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[52] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International Journal of Computer Vision **104** (2013) 154–171

[53] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Proceedings of European Conference on Computer Vision. (2014) 297–312

[54] Zhu, Y., Urtasun, R., Salakhutdinov, R., Fidler, S.: segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4703–4711

[55] Zhang, Y., Sohn, K., Villegas, R., Pan, G., Lee, H.: Improving object detection with deep convolutional networks via bayesian optimization and

structured prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 249–258

[56] Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1440–1448

[57] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99

[58] Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Proceedings of European Conference on Computer Vision. (2014) 392–407

[59] Liu, Y., Guo, Y., Wu, S., Lew, M.S.: Deepindex for accurate and efficient image retrieval. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. (2015) 43–50

[60] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2014) 806–813

[61] 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 22nd ACM International Conference on Multimedia, ACM (2014) 157–166

[62] Sun, S., Zhou, W., Li, H., Tian, Q.: Search by detection: Object-level feature for image retrieval. In: Proceedings of International Conference on Internet Multimedia Computing and Service. (2014) 46

[63] Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Proceedings of European Conference on Computer Vision. (2014) 584–599

[64] Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: A baseline for visual instance retrieval with deep convolutional networks. arXiv preprint arXiv:1412.6574 (2014)

[65] Babenko, A., Lempitsky, V.: Aggregating local deep features for image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1269–1277

[66] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1529–1537

[67] Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1377–1385

[68] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440

[69] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)

[70] Lin, G., Shen, C., Reid, I., et al.: Efficient piecewise training of deep structured models for semantic segmentation. arXiv preprint arXiv:1504.01013 (2015)

[71] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1653–1660

[72] Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5325–5334

[73] Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: Advances in Neural Information Processing Systems. (2014) 1736–1744

[74] Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems. (2014) 1799–1807

[75] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 648–656

[76] Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1347–1355

[77] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2006) 2169–2178

[78] Ramesh, B., Xiang, C., Lee, T.H.: Shape classification using invariant features and contextual information in the bag-of-words model. Pattern Recognition **48** (2015) 894–906

[79] Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. International Journal of Computer Vision **66** (2006) 231–259

[80] Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing **74** (2011) 3823–3831

[81] Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. International Journal of Computer Vision **94** (2011) 335–360

[82] Thomee, B., Lew, M.S.: Interactive search in image retrieval: a survey. International Journal of Multimedia Information Retrieval **1** (2012) 71–86

[83] Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M.: High-level event recognition in unconstrained videos. International Journal of Multimedia Information Retrieval **2** (2013) 73–101

[84] Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M.: A comprehensive performance evaluation of 3D local feature descriptors. International Journal of Computer Vision (2015) 1–24

[85] Buoncompagni, S., Maio, D., Maltoni, D., Papi, S.: Saliency-based keypoint selection for fast object detection and matching. Pattern Recognition Letters (2015)

[86] Lin, W.C., Tsai, C.F., Chen, Z.Y., Ke, S.W.: Keypoint selection for efficient bag-of-words feature generation and effective image classification. Information Sciences **329** (2016) 33–51

[87] Wu, S., Lew, M.S.: Evaluation of salient point methods. In: Proceedings of the 21st ACM International Conference on Multimedia. (2013) 685–688

[88] Wu, S., Lew, M.S.: Salient features for visual word based image copy detection. In: Proceedings of International Conference on Multimedia Retrieval. (2014) 475–478

[89] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395

[90] Yu, G., Morel, J.M.: A fully affine invariant image comparison method. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. (2009) 1597–1600

[91] Moravec, H.P.: Towards automatic visual obstacle avoidance. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence. (1977) 584

[92] Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference. Volume 15. (1988) 50

[93] Beaudet, P.R.: Rotationally invariant image operators. In: International Joint Conference on Pattern Recognition. Volume 579. (1978) 583

[94] Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of the 8th International Conference on Computer Vision. Volume 1. (2001) 525–531

[95] Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision **60** (2004) 63–86

[96] Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Proceedings of European Conference on Computer Vision. (2006) 430–443

[97] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Proceedings of European Conference on Computer Vision. (2002) 128–142

[98] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767

[99] Tuytelaars, T., Van Gool, L.J.: Content-based image retrieval based on local affinely invariant regions. In: Visual Information and Information Systems. (1999) 493–500

[100] Tuytelaars, T., Van Gool, L.J.: Wide baseline stereo matching based on local, affinely invariant regions. In: Proceedings of the British Machine Vision Conference. (2000) 412–422

[101] Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2004) 506–513

[102] Van De Sande, K.E., Gevers, T., Snoek, C.G.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1582–1596

[103] Li, B., Xiao, R., Li, Z., Cai, R., Lu, B.L., Zhang, L.: Rank-SIFT: Learning to rank repeatable local interest points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 1737–1744

[104] Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 2911–2918

[105] Dong, J., Soatto, S.: Domain-size pooling in local descriptors: DSP-SIFT. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5097–5106

[106] Alcantarilla, P.F., Bartoli, A., Davison, A.J.: KAZE features. In: Proceedings of European Conference on Computer Vision. (2012) 214–227

[107] Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision **37** (2000) 151–172

[108] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630

[109] Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: Proceedings of the 21st International Conference on Pattern Recognition. (2012) 2681–2684

[110] Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Proceedings of European Conference on Computer Vision. (2012) 759–773

[111] Figat, J., Kornuta, T., Kasprzak, W.: Performance evaluation of binary descriptors of local features. In: Computer Vision and Graphics. (2014) 187–194

[112] Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. International Journal of Computer Vision **73** (2007) 263–284

[113] Mukherjee, D., Wu, Q.J., Wang, G.: A comparative experimental study of image feature detectors and descriptors. Machine Vision and Applications **26** (2015) 443–466

[114] Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 9–16

[115] Agrawal, M., Konolige, K., Blas, M.R.: Censure: Center surround extremas for realtime feature detection and matching. In: Proceedings of European Conference on Computer Vision. (2008) 102–115

[116] Shi, J., Tomasi, C.: Good features to track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (1994) 593–600

[117] Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. In: Proceedings of European Conference on Computer Vision. (2010) 778–792

[118] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision. (2011) 2564–2571

[119] Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Proceedings of European Conference on Computer Vision. (2010) 183–196

[120] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Robotica **23** (2005) 271–271

[121] Fischer, P., Dosovitskiy, A., Brox, T.: Descriptor matching with convolutional neural networks: a comparison to SIFT. arXiv preprint arXiv:1405.5769 (2014)

[122] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8

[123] Trzcinski, T., Christoudias, M., Fua, P., Lepetit, V.: Boosting binary keypoint descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2874–2881

[124] Levi, G., Hassner, T.: LATCH: learned arrangements of three patch codes. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. (2016) 1–9

[125] Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 815–830

[126] Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 524–531

[127] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010) 3304–3311

[128] 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

[129] Thomee, B., Bakker, E.M., Lew, M.S.: TOP-SURF: a visual words toolkit. In: Proceedings of the International Conference on Multimedia. (2010) 1473–1476

[130] Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. International Journal of Computer Vision **87** (2010) 316–336

[131] Grana, C., Borghesani, D., Manfredi, M., Cucchiara, R.: A fast approach for integrating orb descriptors in the bag of words model. In: IS&T/SPIE Electronic Imaging. Volume 8667. (2013) 866709–1–866709–8

[132] O'Hara, S., Draper, B., et al.: Are you using the right approximate nearest neighbor algorithm? In: IEEE Workshop on Applications of Computer Vision. (2013) 9–14

[133] Salton, G., McGill, M.J.: Introduction to modern information retrieval. London: Library Association Publishing (1986)

[134] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88** (2010) 303–338

[135] Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)

[136] Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In: Proceedings of the International Conference on Multimedia Information Retrieval. (2010) 527–536

[137] Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 817–824

[138] Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science. (2006) 459–468

[139] Heo, J.P., Lee, Y., He, J., Chang, S.F., Yoon, S.E.: Spherical hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 2957–2964

[140] Jin, Z., Li, C., Lin, Y., Cai, D.: Density sensitive hashing. IEEE Transactions on Cybernetics, **44** (2014) 1362–1371

[141] Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3270–3278

[142] Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1556–1564

[143] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1717–1724

[144] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)

[145] Sicre, R., Jurie, F.: Discriminative part model for visual recognition. Computer Vision and Image Understanding **141** (2015) 28–37

[146] Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014)

[147] Azizpour, H., Razavian, A., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2015) 36–45

[148] Zheng, L., Wang, S., Tian, L., He, F., Liu, Z., Tian, Q.: Query-adaptive late fusion for image search and person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1741–1750

[149] Ng, J., Yang, F., Davis, L.: Exploiting local features from deep networks for image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2015) 53–61

[150] Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)

[151] Nandakumar, K., Chen, Y., Dass, S.C., Jain, A.K.: Likelihood ratio-based biometric score fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 342–347

[152] Zhang, S., Yang, M., Cour, T., Yu, K., Metaxas, D.N.: Query specific fusion for image retrieval. In: Proceedings of European Conference on Computer Vision. (2012) 660–673

[153] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. (2014) 675–678

[154] Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proceedings of European Conference on Computer Vision. (2008) 304–317

[155] Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision **42** (2001) 145–175

[156] Zheng, L., Wang, S., Liu, Z., Tian, Q.: Packing and padding: Coupled multi-index for accurate image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1939–1946

[157] Spyromitros-Xioufis, E., Papadopoulos, S., Kompatsiaris, I.Y., Tsoumakas, G., Vlahavas, I.: A comprehensive study over vlad and product quantization in large-scale image retrieval. IEEE Transactions on Multimedia **16** (2014) 1713–1728

[158] Jégou, H., Zisserman, A.: Triangulation embedding and democratic aggregation for image search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3310–3317

[159] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proceedings of European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision. Volume 1. (2004) 1–2

[160] Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory. (1992) 144–152

[161] Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., Huang, T.: Large-scale image classification: fast feature extraction and svm training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 1689–1696

[162] Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Proceedings of European Conference on Computer Vision. (2010) 143–156

[163] Jaakkola, T.S., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. Advances in Neural Information Processing Systems (1999) 487–493

[164] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)

[165] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1026–1034

[166] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings of European Conference on Computer Vision. (2014) 818–833

[167] Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Proceedings of the 18th International Conference on Artificial Intelligence and Statistics. Volume 2. (2015) 6

[168] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **7** (1993) 669–688

[169] Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. (2015) 84–92

[170] Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3476–3483

[171] Yang, S., Ramanan, D.: Multi-scale recognition with dag-cnns. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1215–1223

[172] Ma, L., Lu, J., Feng, J., Zhou, J.: Multiple feature fusion via weighted entropy for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3128–3136

[173] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning. (2011)

[174] Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., Yang, S., Wang, Z., Xiong, Y., Qian, C., et al.: Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. arXiv preprint arXiv:1409.3505 (2014)

English Summary

With the ever-growing amount of image data on the web, much attention has been devoted to large scale image search. It is one of the most challenging problems in computer vision for several reasons. First, it must address various appearance transformations such as changes in perspective, rotation and scale existing in the huge amount of image data. Second, it needs to minimize memory requirements and computational cost when generating image representations. Finally, it needs to construct an efficient index space and a suitable similarity measure to reduce the response time to the users. This thesis aims to provide robust image representations that are less sensitive to above mentioned appearance transformations and are suitable for large scale image retrieval.

Early approaches, the Bag-of-Words (BoW) model and its variants, have dominated the research on large scale image retrieval. The pipeline of BoW for image retrieval mainly consists of three steps: (i) salient point feature extraction; (ii) visual vocabulary generation; (iii) BoW based feature encoding. In each step, many efforts have been made to achieve state-of-the-art performance on large scale image search.

First, we investigated the strengths and weaknesses of the existing salient point detectors and descriptors on diverse image distortions. The comparative experimental studies we presented can support researchers in choosing an appropriate detector and descriptor to generate the BoW based image representation.

Compared to the real valued local descriptors, binary string local descriptors have the advantage of low memory requirements and efficient matching via Hamming distance. We further proposed to use the "K-majority" cluster method with ANN

search to generate a BoW image representation based on binary string local descriptors. The evaluation results showed that the binary string descriptor based BoW model has low memory requirements for vocabulary storage and competitive performance compared with real valued local descriptor based BoW image representation.

Since the existing salient point methods are sensitive to viewpoint or perspective changes, we further proposed a novel salient point descriptor named RIFF. RIFF is generated according to pair-wise intensity comparisons over a sampling pattern inspired by the human retina. The evaluation results showed that the RIFF based BoW image representations outperformed other feature descriptors with respect to invariance to scale, rotation, and viewpoint transformations.

More recently, image representations generated by the convolutional neural networks (CNNs) have demonstrated their high performance compared to the state-of-the-art for image retrieval. In this thesis, we explored both real valued and binary string image representations based on feature maps from the layers within CNNs. In addition, we presented a fusion scheme to further improve image search accuracy. Moreover, we designed a more powerful CNN architecture to improve the robustness of CNN models.

Finally, although this thesis makes a substantial number of contributions to large scale image retrieval, we also presented additional challenges and future research based on the contributions in this thesis.

Nederlandse Samenvatting

Met de al maar groeiende hoeveelheid visuele data op het web is er veel aandacht besteed aan het grootschalig zoeken naar afbeeldingen. Er zijn verschillende redenen waarom dit een van de grote uitdagingen op het gebied van computer vision is. Allereerst moet er rekening gehouden worden met visuele transformaties zoals perspectief, rotatie en schaling die zijn toegepast op afbeeldingen uit deze enorme hoeveelheid data. Ten tweede is het noodzakelijk de vereiste hoeveelheid geheugen en rekenkracht te minimaliseren die bij het genereren van de afbeeldingsrepresentaties nodig is. Tenslotte moet er een efficiënte index-ruimte en een geschikte afstands maat gecreëerd worden om wachttijd van de gebruiker te verkleinen. Deze scriptie beoogt robuuste afbeeldingsrepresentaties te leveren die minder gevoelig zijn voor de hiervoor genoemde visuele transformaties en geschikt zijn voor het grootschalig zoeken naar afbeeldingen.

Al langer bestaande technieken zoals het Bag-of-Words (BoW) model en haar varianten domineren het onderzoek naar grootschalig zoeken naar afbeeldingen. Het process van BoW bestaat voornamelijk uit drie stappen: (i) salient point feature extraction, (ii) visual vacabulary generation, (iii) het op BoW gebaseerde feature encoding. Veel onderzoek heeft zich gericht op het bereiken van de state-of-the-art prestaties voor grootschalig zoeken naar afbeeldingen.

Als een eerste stap onderzoeken we de sterke en zwakke punten van de bestaande salient point detectoren en descriptoren op diverse verdraaiingen van afbeeldingen. De gepresenteerde vergelijkende experimenten kunnen onderzoekers bijstaan in het selecteren van een passende detector en descriptor voor het genereren van een op BoW gebaseerde representatie van afbeeldingen.

NEDERLANDSE SAMENVATTING

Vergeleken met locale descriptoren van reële getallen hebben locale discriptoren bestaande uit een binaire getallenreeks het voordeel weinig geheugen nodig te hebben en efficiënt vergeleken te kunnen worden met behulp van de Hamming-afstand. Verder stellen we voor om de "K-majority" cluster methode met ANN-search te gebruiken voor het genereren van BoW afbeeldingsrepresentaties gebaseerd op locale descriptoren van binaire getallenreeksen. De evaluatie-resultaten laten zien dat deze aanpak weinig geheugen gebruikt voor vocabulaire opslag en competetief presteert vergeleken met BoW afbeeldingsrepresentaties gebaseerd op locale descriptoren van reële getallen.

Aangezien bestaande salient point methodes gevoelig zijn voor de kijkhoek en veranderingen in perspectief stellen we ook een nieuwe salient point descriptor voor, RIFF genaamd. RIFF wordt gegenereerd op basis van paarsgewijze intensiteitsvergelijkingen tussen selectie-patronen geïnspireerd door de patronen van het menselijke netvlies. De evaluatie hiervan laat zien dat de op RIFF gebaseerde BoW afbeeldingsrepresentaties andere kenmerkbeschrijvingen met betrekking tot schaal, rotatie en kijkhoek transformaties weet te overtreffen.

Recentelijk hebben afbeeldingsrepresentaties gegeneerd aan de hand van convolutional neural networks (CNNs) laten zien goed te presenteren ten opzichte van de state-of-the-art als het gaat om het zoeken naar afbeeldingen. In deze scriptie onderzoeken we reëel-waardige en binaire representaties van afbeeldingen gebaseerd op feature maps uit de lagen van de CNNs, en presenteren we een fusie-schema om de nauwkeurigheid bij het zoeken naar afbeeldingen verder te verbeteren. Daarnaast ontwerpen we een krachtigere CNN-architectuur om de robuustheid van CNN modellen te verbeteren.

Tenslotte, ondanks de substantiële bijdragen van deze scriptie op het gebied van het grootschalig zoeken naar afbeeldingen, presenteren we verdere uitdagingen en nieuwe onderzoeksrichtingen naar aanleiding van de bijdragen uit dit proefschrift.

Acknowledgements

First of all, I would like to thank the collaborations with Dr. Erwin M. Bakker, Dr. Bart Thomée, and Dr. Ard Oerlemans for some publications. Thank you very much for teaching me the art of scientific writing. I want to thank Dr. Fons Verbeek and Dr. Erwin M. Bakker for the post-doc position recommendation. I also want to thank my master supervisor Prof. Guoqiang Xiao (Southwest University, China) for the help of my research and career planning.

Second, I would like to thank my colleagues in the media lab, LIACS, Yanming Guo and Yu Liu, thank you for organizing the research seminar every week and introducing me to the background of deep learning. I also want to thank Prof. Feng Yao (National University of Defense Technology, China) for good suggestions about my research.

I am deeply grateful for my dear parents, you raised me, taught me how to be good, have always been supportive and meticulous caring. I wish we can spend more time together and travel a lot in the near future. I also want to thank my girlfriend Cailing Tang. Even as we face difficulties due to our long distance relationship, we always trust the happy future.

Many thanks to my friends in the Netherlands: Zhan Xiong, Junlin He, Boyang Liu, Qinyin Hu, Shengfa Miao, Di liu, Fuyu Cai, Yuanhao Guo, Xiaoqing Tang, Enrique Larios Vargas, Zhiwei Yang, Kaifeng Yang, Channa Li, Hao Wang, Longmei Li, Koen van der Blom, Minghao Li, Peng Wang, Hongchang Shan, Jian Yang, Jinxian Wang, Yuchuan Qiao, Meng Sun, Min He, Guangchao Chen, Yinlong Xiao, Zhenyu Xiao, Yaojin Pen, Wenbo Ma, Rui Zhang, and Feng Zhang.

ACKNOWLEDGEMENTS

Many thanks to my friends in China: Han Jiang, Wei Xiong, Heng Yang, Fei Yuan, Zhen Tan, Chong Chen, Yu Guan, Kaifeng Deng, Fan Jiang, Feiwu Yuan, and Xiao Lin.

Curriculum Vitae

Song Wu was born in Mianyang, Sichuan, China on September 13, 1985. He received the B.S. degree and the M.S degree (under the supervision of Prof. Guoqiang Xiao) of computer science from Southwest University, Chongqing, China, in 2009 and 2012, respectively.

In September 2012, he obtained a PhD position at the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands, under the supervision of Prof. Dr. J.N. Kok and Dr. M.S. Lew. His research mainly focuses on image pattern recognition, feature descriptors and detectors, large scale image search and classification, machine learning and deep learning. He is a reviewer of of the International Journal of Multimedia Information Retrieval (2012-now) and the International Conference of British Machine Vision Conference (BMVC 2014-now).