Deep Learning with Query Sensitive Attention Mechanisms for Content-based Image Retrieval

Zechao Hu

Doctor of Philosophy

University of York
Computer Science
September 2022

Abstract

Content-based Image Retrieval (CBIR) is the task of searching for the most similar images to the query content from an extensive image database. Most existing feature extraction methods and attention mechanisms for CBIR tasks are query non-sensitive, ignoring the specifics of the query pattern, which may lead to focusing on irrelevant regions to the query content. In this thesis, we explore query sensitive attention mechanisms for CBIR task, which involves query feature information in the feature extraction procedure of the candidate image.

Firstly, we propose the Conditional Attention Network (CANet). CANet takes the query image and a candidate image as input, resulting in a co-attention map of the candidate image under the condition of the query content. The generated co-attention map could correctly highlight the target object and improve image retrieval performance when embedded into a convolution neural network (CNN) based feature extraction pipeline.

Secondly, another more efficient co-attention method is proposed based on local feature selection and clustering over candidate local features. Using local feature selection and clustering dramatically reduces the computation costs caused by the query sensitivity but still leads to accurate co-attention maps even under challenging situations. The proposed clustering-based co-attention method leads to new state-of-the-art performance on several benchmark datasets.

Lastly, we explore using clustered expressive local features to perform many-to-many local feature matching for CBIR. We show that the proposed local feature matching method implicitly generates co-attention-like local matching maps. In addition, a trainable binary encoding layer is applied for network fine-tuning, enabling the model to generate compact binary codes with slight performance degradation and greatly reducing computation costs.

In summary, we demonstrate that the query information could play an important role in feature extraction for the CBIR task. With a simple design, co-attention could be practical and effective even for large-scale image retrieval tasks.

Acknowledgements

I would like to express my appreciation and gratitude to everyone who gave me any kind of help during my 4 years PhD study.

First, I am sincerely grateful to my supervisor: Dr. Adrian G. Bors. Whenever I encounter problems or reach a bottleneck in my research, Dr. Bors always gives thoughtful support and guidance, helping me overcome obstacles during my study. In addition, every time I prepared a paper for publishing, detailed feedback and writing suggestion were received from him, which benefited me a lot and improved my writing skills.

I would also like to thank my internal assessor: Dr William Smith, who not only provides feedback but also gives inspiring advertisements and interesting questions about my research study in our TAP meetings.

My friends: Fei Ye, Guoxi Huang, and Cameron also helped me a lot during my PhD study, and I really enjoyed the time spent with all of them.

Finally, I would like to thank my parents, Ling Zhang and Donghong Hu, who gave full support from both aspects of physically and mentally for my PhD study.

Declaration

I declare this thesis is composed solely by myself, and all the contributions presented in this thesis result from my own work. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references. The contents of some chapters have been published in the following:

- Hu, Zechao, and Adrian G. Bors. "Conditional attention for content-based image retrieval." British Machine Vision Conference (BMVC) 2020.
- Hu, Zechao, and Adrian G. Bors. "Expressive Local Feature Match for Image Search." IEEE International Conference on Pattern Recognition (ICPR) 2022.
- Hu, Zechao, and Adrian G. Bors. "Enabling large-scale image search with co-attention mechanism." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023.
- Hu, Zechao, and Adrian G. Bors. "Few but informative local hash code matching for image retrieval." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023.

Contents

List of Figures

Chapter 1

Introduction

The development of modern technology has made photo-capture devices, such as cameras and smart mobile phones, more accessible and widely used in different aspects of life. Many image data are generated following various activities, such as social media, medical, industrial, educational, and others. Especially, with the advancement of the Internet, many images are produced, stored, and spread worldwide every day. All these human activities require a system to organize images so users can easily find them. Generally, image retrieval represents the task of searching and retrieving images, which would semantically match the query input, from a large database of digital images. The earliest image retrieval work could track back to the conference: Database Techniques for Pictorial Applications [12] in the 1970s, after which image retrieval started to attract interest. As shown in Figure 1.1, we can talk about different types of image retrieval systems depending on the query format.

Initially, the most common image retrieval system would be text-based image retrieval (TBIR). TBIR system utilizes textual image annotations, such as image file names, tags, keywords, descriptions, or even GPS coordinates, to search desired images. TBIR is computationally efficient as storing textual annotation for each image takes minimal space and the text string matching is fast at the retrieval stage. However, it suffers from several disadvantages. First, it is not feasible to manually annotate a large-scale image database with descriptive texts. Second, the manually added textual annotations only reflect the annotator's understanding of the image, making it subject to individual human perception. In the worst case, if the end-user and the annotator have a different understanding

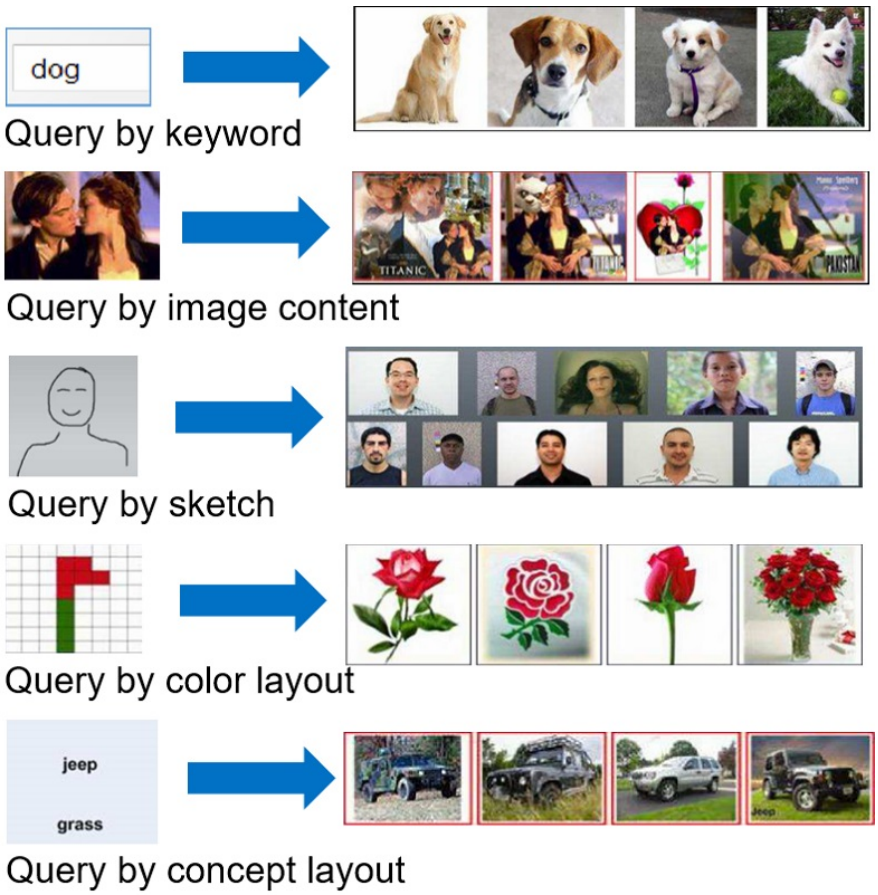


Figure 1.1: Illustration of different query formats and corresponding results. Images taken from [156].

of the image, a TBIR system will likely return irrelevant results. Besides, the textual annotation may not be able to comprehensively describe all content of the image, especially when there are multiple objects in a single image. Finally, the text-based approach is also likely to be restricted by language. For example, if the user language changes, pre-annotated descriptions would not work. To overcome these limitations of TBIR, an alternative way of image retrieval, called content-based image retrieval, was proposed.

Content-based image retrieval (CBIR) is also known as Query By Image Content (QBIC). CBIR is a more intuitive and user-friendly way of image retrieval as it does not require written text information but directly takes an image as input, returning a set of similar ones to the query content. In addition, directly utilizing image intrinsic information instead of textual annotation makes the image retrieval system more practical, as the text description may be inaccurate or not even related in any way to the content of the images.

Apart from traditional TBIR and CBIR, some specialized query formats have also been

developed in the image retrieval field. For instance, sketch-based image retrieval could be treated as a related approach to sample-based image retrieval. Sketch-based approaches allow the user to search images that would match the semantics corresponding to the intuition from the user's mind. It could be helpful, especially when the user's search intention is not well defined and there is no access to a proper sample image as query item [16]. Although initial sketch-based image retrieval methods only work well for some simple patterns or specific artworks [156], in some later research, such as the Edgel [15], it is used with natural images. However, there are some non-trivial problems with the sketch-based image retrieval approach. The major one could be that users may not be able to draw the contour during the actual application quickly.

Another type of query format is the colour layout. It searches images that match the colour spatial co-relation in different image regions. This approach provides the search intention of the user in some specific situations, but it suffers from limited semantic meaning that can be expressed by using only colour and layout information. Moreover, significant changes in the illumination conditions or the image acquisition parameters could also present a major challenge for this colour-based approach. Similar to the colour layout format, the concept layout query format from [142] proposes to combine spatial layout information with text information for image retrieval. This query format allows the user to search for a more complex semantic representation and may be suitable for particular specialized query intentions. However, such an approach requires that the image database is pre-processed using an object recognition algorithm, annotating the object class and spatial location.

Specialized CBIR has been employed in various applications such as medical images, satellite images, remote sensing and others. Specialized image retrieval relies on identifying and using specific features characteristic to the modalities of that domain. This thesis focuses on content-based image retrieval (CBIR) in generic images. Lately, landmark datasets, such as ROxford and RParis [96] that contain photos of landmark buildings taken under various situations and conditions have become a prevalent CBIR research task and application scenario in recent CBIR works [96]. Content-based image retrieval in this thesis refers to retrieving images that contain the same object or content as the query image. In other CBIR works, such as [128], the goal is to retrieve images from the same class as those shown in the query image. Recent state-of-the-art works of CBIR

[14, 123, 146] focus on the same object (or content) retrieval and the work in this thesis follows this practice.

A general pipeline of content-based image retrieval is illustrated in Figure. 1.2. It consists of two stages: the offline stage and the online stage. The offline stage mainly involves feature extraction and caching for each candidate image from the image database. After feature extraction, each candidate image will be mapped from the original RGB colour pixels to compact feature vector representations. The candidate image feature extraction would only need to be done once and extracted features will be cached permanently for usage in the future online stage. Thus, it has memory usage limitations for the offline stage. During the online stage, the query image is processed by the same feature extraction module used for the database images. Then a similarity measure is employed between the features of the query and each database' image. If the feature extraction results in some real-value feature vectors, the similarity measure can use the L2 distance (cosine similarity). When the image representation consists of a binary code, then the Hamming distance can be used. Moreover, some other works would extend the usage of deep learning as the similarity evaluation module. Instead of applying such classic distance metric as mentioned above, these works utilize a trainable layer, like a fully connected layer, to directly learn a distance metric for image match [150, 34]. The processing time is essential for the online stage module of the CBIR.

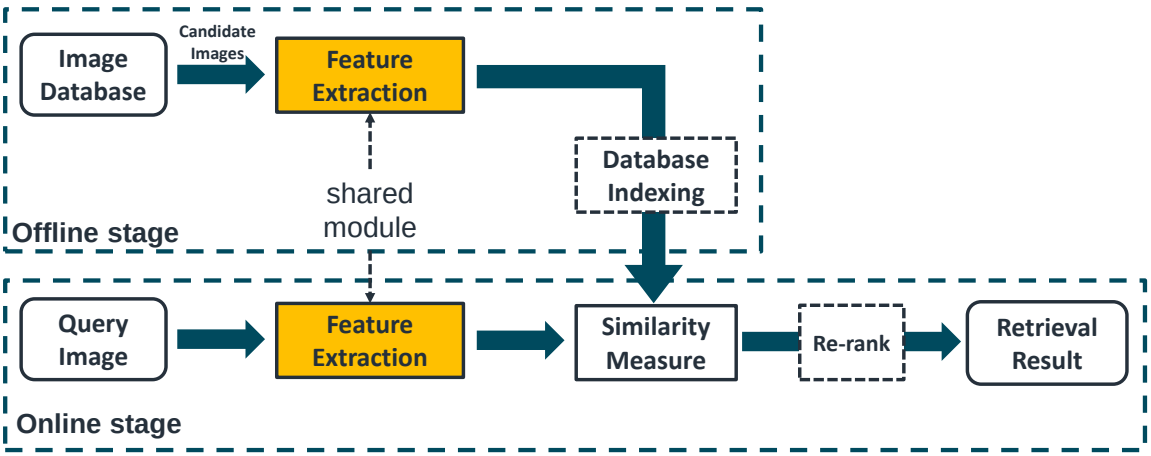


Figure 1.2: Illustration of content-based image retrieval pipeline.

To further improve the performance of the CBIR pipeline, there are two optional modules: database indexing and the re-ranking module, marked with dash-line rectangles in Figure 1.2. Database indexing refers to the organization of the image database structure

to improve the retrieval outcome. A database indexing method could significantly decrease the response time of the retrieval system and it plays a more and more critical role as the database gets large. For example, the inverted file indexing has been successfully implemented to speed up image retrieval [112]. Initially, inverted file indexing [159] was a method implemented in text processing that can map the content to its location within the documents or database files. Inverted file indexing is adapted for image retrieval to help the retrieval system quickly pick out candidate images that share some common visual words with the query image [112]. This procedure reduces the number of candidate images compared with the query, resulting in a more efficient response.

The re-ranking module aims to refine the initial retrieval results with some extra feature information or processing procedure. It could be based on different techniques, such as spatial verification [87] or query expansion [26]. For instance, query expansion, also adapted from text retrieval, consists of reissuing the top rank retrieval results to generate new queries. Because some relevant features or information may not be included with the original query image, generating new queries and then using them for initial result refinement can reduce the omission of related image retrieval.

Due to the variation of image content, crucial components within a CBIR pipeline are still the feature extraction procedure and the corresponding similarity measure. Therefore, most existing studies are also carried out around these two modules to improve image retrieval performance. Early CBIR systems would employ hand-crafted feature extractors using low-level features such as colour, texture features or characterizing the gradient information to build compact feature representations for the input images. However, due to the difficulty of bridging the gap between low-level feature information and high-level semantic meaning, such hand-crafted approaches eventually reach their bottleneck. Nevertheless, lately, the CBIR field was revolutionized by deep learning.

Deep learning uses huge image databases to train neural networks with multiple layers for extracting complex features from images. The layer-wise operation within a deep learning model would automatically extract features in different semantic levels from the different layers. Thus, with proper structure and training, the deep learning model could implicitly learn complex feature extraction functions without relying on low-level feature information or domain knowledge from the algorithm designer. To be more specific, there are several types of deep model structures used in deep learning, such as deep neural

network (DNN), deep belief network [63] and convolutional neural network (CNN). Among these deep models, due to attributes like parameter sharing and translation-invariance, CNNs serve as a common backbone network structure choice for many deep learning works [132, 144, 147, 155, 157] and exhibit outstanding performance in terms of both accuracy and computation. In recent CBIR works [85, 138, 87, 14, 146, 80, 145], combining with proper attention mechanism could further refine the feature output provided by the CNN while improving the whole model's retrieval performance. Currently, these CNN-based works provide the newest state-of-the-art retrieval performance in major benchmark datasets for content-based image retrieval.

However, most existing attention mechanisms [85, 138, 87, 14, 146, 80, 145] for CBIR are query non-sensitive: they take single candidate images as input and predict the region of interest purely based on the knowledge learned during the training, regardless of what the query content is. This kind of attention mechanism tends to highlight the relevant objects or regions from the image uniformly. As a result, the CBIR system could look for regions outside the actual object of interest and is likely to fail. It happens especially when the target object is not salient or is surrounded by distractors related to the training data. Figure 1.3 shows some examples in which the query non-sensitive attention mechanism from WGeM [138] fails. WGeM is a CBIR model trained on a landmark building dataset: rSfM-120k [97, 98]. The Louvre Palace and the Louvre Pyramid building in images of Figure 1.3 are both training data relevant and could be potential objects of interest. It can be observed that when considering the Louvre Pyramid as the query, which is semi-transparent and textured in Figure 1.3 (a), it is always ignored by the attention module. At the same time, the adjacent building (the Louvre Palace) attracts the most attention, as observed in Figure 1.3 (d).

Ideally, the attention should be query sensitive: it is supposed to be consistent with the query content. For example, as shown in Figure 1.4, if we use the Louvre Pyramid as the query item, the pyramid part should be highlighted, while if the Louvre Palace is the query item, then the palace building should be highlighted. This kind of query sensitive attention that is conditioned on the query content is called co-attention in this thesis.

For the CBIR task, the intuition of applying co-attention is that, given the query image, we would pay more attention to regions from the target image which is similar to the query content. For example, if the query image is a tower building, people would only care about

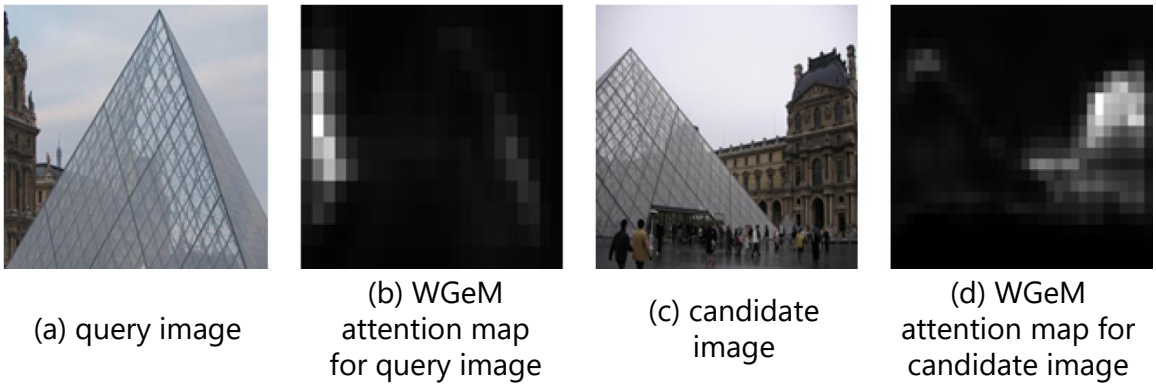


Figure 1.3: An example in which the query non-sensitive, trainable convolution layer based attention module from WGeM [138] fails. The examples show the query (a) and its WGeM attention map (b) together with the search image and its corresponding WGeM attention map. Images are taken from [138].

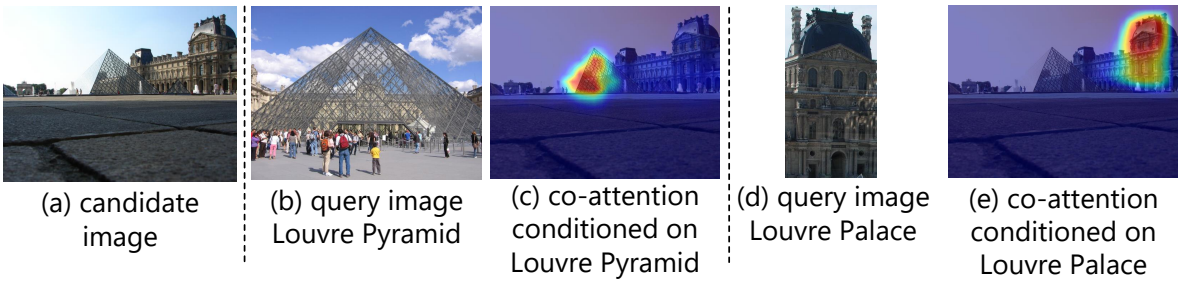


Figure 1.4: Examples of query sensitive co-attention maps. Image (a) shows the candidate image. (c), (e) shows the co-attention map conditioned on query images (b) and (d) separately.

tower-like content from the candidate image. If the candidate image contains several potentially relevant objects, people would compare each of them to the query. If any of them matches with the query, then only this region is supposed to be highlighted and the candidate image should be treated as a positive match. In other words, the motivation for using co-attention mechanisms is to generate attention maps that are dynamically conditioned by the input query content for the candidate image, leading to better relevant feature selection or re-weighting for CBIR. Actually, in some other computer vision tasks [46, 83, 130], the query pattern has shown to be essential for feature extraction and image recognition task.

Due to its query sensitivity, the co-attention mechanism could cause high extra computation costs for a CBIR system, from both aspects of memory usage at the offline stage and time consuming at the online stage. On the one hand, as the attention score of each candidate image local feature remains unknown before seeing the actual query image

at the online retrieval stage, all potentially useful local features of the candidate image need to be cached. It will cause much more memory usage than the query non-sensitive global feature methods, which would only need to cache a compact feature vector for each candidate image. On the other hand, the co-attention map generation and the final co-attention weighted feature vector building of each candidate image must be performed before the similarity measure at the online stage. Consequently, considering a complex co-attention generation procedure would cause extra time costs for the online retrieval procedure. These extra computation costs could be unaffordable and make co-attention impractical, especially for large-scale image retrieval. The work of this thesis focuses on improving the performance of the CNN-base image retrieval pipeline by embedding effective and efficient co-attention mechanisms into the feature extraction procedure.

1.1 List of contributions

The main contributions of this thesis are listed as follows:

- First, in Chapter 3, we propose a query-sensitive model, namely the Conditional Attention Network (CANet), for localizing the object (region) of interest from the candidate image that matches the content of the query image. A key-point based region-level annotation generation pipeline is also provided. Thus, the whole model is end-to-end trained in a self-supervised manner and can be combined with generic CNN-based feature extraction methods to boost the original CBIR method's retrieval performance.

- The second work in Chapter 4 proposes a more efficient clustering-based co-attention method which greatly relieves the extra computation cost problem caused by query sensitivity, making it practical in large-scale CBIR tasks and reaches new state-of-the-art results with benchmark datasets.

- Based on the clustering-based feature extraction strategy from the former chapter, in Chapter 5, we propose an effective many-to-many local match method applied with those few but expressive clustered local features for image retrieval. Furthermore, a trainable binary encoding layer is additionally embedded into the feature extraction pipeline, which further significantly reduces the computation cost but still can

generate interpretable co-attention-like local match maps with slight performance degradation.

1.2 Thesis outline

The rest of this thesis is organized as follows:

- Chapter 2 presents related works to the CBIR task, including the description of major conventional image retrieval methods, deep convolutional neural networks and deep learning-based CBIR methods. In addition, the achievements and drawbacks of existing works are summarized and discussed.

- Chapter 3 represents our first research work: Conditional Attention Network (CANet). The proposed CANet is a self-supervised co-attention network that aims to generate query-sensitive attention maps for CNN-based image feature re-weighting, leading to better retrieval results. The core component of CANet is the multi-scale convolution block, which is designed to fuse query image global feature and candidate image local features. In addition, a SuperPoint [30] based data generation pipeline is also proposed, which can automatically detect region-level correspondence between existing matching image pairs from the rSfM-120k [98]. These detected matching regions serve as the training data for CANet.

  Hu, Zechao, and Adrian Gheorghe Bors. "Conditional attention for content-based image retrieval." British Machine Vision Conference (BMVC) 2020.

- In Chapter 4, we propose an improved non-trainable-parameter co-attention method for large-scale image retrieval. The proposed co-attention method is based on L2-norm feature selection and local feature clustering, which significantly relieves the extra computation cost caused by the co-attention. Still, it can generate good co-attention even under some challenging situations. Moreover, for quantitative retrieval results, the proposed co-attention method dramatically improves the original baseline model performance and achieves new state-of-the-art results on common benchmark datasets.

  Hu, Zechao, and Adrian G. Bors. "Enabling large-scale image search with co-

attention mechanism." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023.

- Chapter 5 proposes an expressive local feature match method for CBIR. It is also based on L2-norm feature selection and local feature clustering. Instead of generating co-attention for feature re-weighting, it explores performing local feature matching in a many-to-many manner for image retrieval. Additional binary encoding and fine-tuning techniques are embedded, significantly reducing the computation cost at the retrieval stage with only a slight performance deterioration. Experimental results also demonstrate that with much lower computation requirements, the local matching method could generate high-quality co-attention-like matching maps between pairs of images.

  Hu, Zechao, and Adrian G. Bors. "Expressive Local Feature Match for Image Search." IEEE International Conference on Pattern Recognition (ICPR) 2022.

  Hu, Zechao, and Adrian G. Bors. "Few but informative local hash code matching for image retrieval." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023

- Chapter 6, the conclusion, explains the connection and relation among the research studies from the three research chapters while also providing a summary. Possible future research directions are also discussed.

Chapter 2

Related work

In this chapter, related works to the CBIR task are reviewed. As the feature extraction module is the central module of the image retrieval pipeline, the literature review starts with some early conventional hand-crafted feature extraction methods. These early stage works lay a solid foundation for image retrieval research. Some classic ideas or module designs still play an important role in recent deep learning based CBIR works. Then, it follows by deep learning based feature extraction methods. After that, in a separate section, we discuss existing attention mechanisms, which can serve as an important add-one module for feature selection and re-weighting in recent CBIR works. Finally, some additional works about database indexing and re-ranking technology that also contributes to CBIR system performance are introduced.

2.1 Conventional feature method

In this section, some representative conventional CBIR methods are reviewed. First, some early hand-crafted feature extractors based on different types of low-level feature information are introduced. These approaches aim to transform raw input pixel images into compact feature representations for efficient image retrieval. Then, some more complex feature aggregation methods are illustrated. These feature aggregation methods could be treated as a separate post-processing module, taking pre-extracted local features as input while building a more meaningful and compact representation for CBIR.

2.1.1 Low-level features

As mentioned in Chapter 1, early stage conventional CBIR methods are commonly based on hand-crafted feature extractors, using low-level feature information, such as colour, texture, shape or gradient. In the following, some representative works corresponding to each category of features are discussed.

Color

As one of the most extensive vision characteristics of colour images, colour information is invariant to changes in multiple image acquisition parameters, such as scale, translation, rotation or minor changes in camera viewpoint. It has been widely used in early image or object search works. The colour histogram [117] is one of the earliest and most popular colour information based image representation methods for image comparison. The colour histogram represents an empirical probability density function of the colour distribution. The size of each colour bin is used to represent the global feature vector to describe the image. The work in [117] demonstrates the effective role that colour histogram based image representation can play in the object identifying task. It also proposes histogram interaction for similarity measure and histogram back-projection for target object localization, making colour histogram based image indexing more practical and interpretable for large-scale image search. However, simple colour histograms can only provide very limited characteristics of the input image. It is likely to fail when two images have similar colour distribution but different content. To improve the colour histogram representation, the colour coherence vector (CCV) [91] embeds the spatial coherence information into the colour histogram feature vector by partitioning each colour bin into coherent and incoherent. The distance calculation is conducted only between bins with the same coherence status. The colour correlograms [48] further improve the colour histogram representation by considering spatial co-relation (distance) changes between colour pairs when building the image representation. Apart from these global histogram based methods, there are some other forms of colour representation, such as colour moments [49] and colour co-occurrence matrix [95]. However, despite the success of the application of colour information for image retrieval, purely colour-based image representation still suffers from limited spatial information and lack of perceptual similarities [3].

Texture

Texture features could be defined by overall image information representation considering colour, shape, image structure, randomness, granularity, linearity, roughness, and homogeneity [3]. Since texture characteristics are also presented in most real-world images, many texture-based feature extraction methods have been proposed for compact image representation building. The Gabor wavelet features [78] could be one of the earliest works that utilize texture as an image feature for image retrieval. The Gabor wavelet features are extracted by a set of Gabor filters with different orientation settings. Each Gabor filter could be treated as an analyzer or feature extractor for specific image content patterns. The final compact global feature vector is constructed by concatenating feature components from all Gabor filters' output. Apart from Gabor features, there are also many other texture-based feature extraction methods or algorithms, such as edge histogram descriptor (EHD) [90], Discrete Wavelet Transform [2]. In general, texture feature extraction methods tend to capture specific patterns from the input image. Such methods describe the surface properties of each object as well as its relationship to surrounding regions. However, textures may suffer from noise sensitivity and the computational complexity required for full information representation [3].

Shape

Different from the previously mentioned elementary feature information: colour and texture, shape information is considered to carry strong semantic information [134, 3], as people sometimes can recognize the target object solely based on the contour shape of the object. However, a single shape-based feature suffers from variations in the scale, rotation or even some tiny differences in the object contour [3]. Accordingly, in the content-based image retrieval task, the shape feature normally serves as a complementary to colour and texture feature information[1]. For instance, in [134], a CBIR pipeline, which combines colour, texture and shape feature information, is proposed. The shape information is described by Pseudo-Zernike moments. The similarity scores with colour, texture and shape feature vectors are weighted and summed to get the final match result.

[1]To distinguish with shape-based image retrieval task, we do not consider shape retrieval methods where the query input is a contour.

Gradient

Gradient information here refers to the magnitude and orientation of texture, edges or other features with respect to the neighbourhood. Scale-invariant feature transform (SIFT) [77] could be one of the most popular gradient-based image local feature extractors.

The pipeline of SIFT algorithm consists of 4 main steps:

Step 1: it finds potential interest points by using a difference-of-Gaussian [77] function, which is invariant to image scaling.

Step 2: key-points, derived from those potential interest points of step 1, are localized to sub-pixel accuracy and the unstable ones will be removed.

Step 3: around each key-point, the gradient of the neighbourhood (orientation and magnitude) is calculated and each area is divided into 4 or 8 blocks. Within each block, all pixel's gradient orientation is assigned to 8 directions and a gradient histogram is built, as shown in Figure 2.1.

Step 4: local descriptor (local feature vector) is built for each key-point centred area by concatenating each block's gradient histogram vector.

SIFT can well capture the invariant feature information to rotation and scaling transformation and is robust to illumination change. In addition, as SIFT is based on key-point localization and only extracts local features centred around these key-points, it is also robust to occlusion and clutter. As one of the most successful image local feature extractors, SIFT has been applied in a variety of computer vision tasks, including image retrieval. For instance, in [61], SIFT local features are compressed by Principal Component Analysis (PCA) algorithm and perform well in image retrieval. One main drawback of SIFT is that it is mathematically complicated and computationally heavy due to the gradient calculation of each key-point. To relieve this problem, the Edge-SIFT [153] improves the original SIFT method by extracting binary coded local features based on the binary edge map of the image, making it more applicable for large-scale image retrieval and getting higher accuracy. Moreover, instead of solely relying on the gradient information, the coupled MultiIndex (c-MI) framework [154] proposes to fuse the local colour

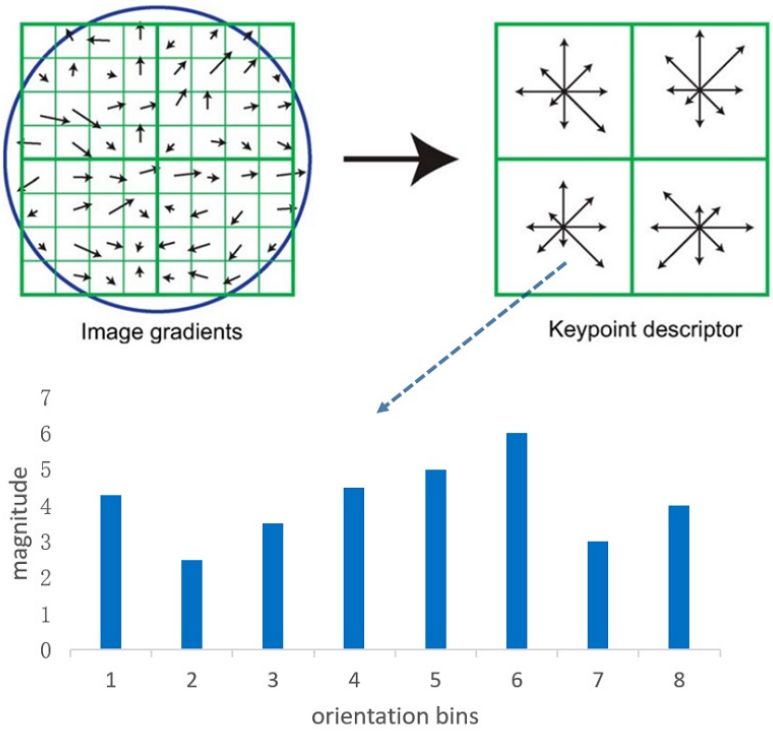


Figure 2.1: Illustration of building gradient orientation histogram from region blocks [77]

feature information into the SIFT features, improving the retrieval accuracy at half of the computational requirements by SIFT.

Inspired by SIFT, another robust image local feature descriptor, Speeded Up Robust Features (SURF) [9], was proposed. SURF describes the intensity content of each detected key-point along with its neighbouring regions by the distribution of first-order Haar wavelet responses instead of the simple magnitude and orientation as in SIFT. In addition, SURF represents the whole image with only 64 dimension feature vectors. This approach speeds up the feature extraction and matching procedure.

2.1.2 Feature aggregation

Apart from methods like those mentioned above that focus on extracting and encoding low-level feature information from raw input images into compact feature vectors, some other works turn to feature aggregation: how to more effectively utilize extracted feature vectors to build more comprehensive image representation form as well as studying the similarity measure to be used for retrieval. These methods normally work with a predefined local feature extractor and serve as a post-processing module to generate better

compact image representations from pre-extracted local features.

The bag of visual words (BoV) [112][2] could be one of the most successful and representative feature aggregation methods. This idea is adapted from the text-processing method called the bag of words (BoW). In text processing, bag of words treats each document as a set of unordered keywords and uses the frequency of each keyword to represent each document. In the bag of visual words [112], it also uses the frequency of "words" to represent each image, while the "words" here means image region characteristic features or the local image descriptor. The bag of visual word pipeline is illustrated in Figure 2.2. At the training stage, as shown in the top part of Figure 2.2, with a set of sample images, a local feature extractor (like SIFT [77]) is implemented to get local descriptors for each image. Then, k-means clustering is applied over these local descriptors, resulting in the cluster centres. These cluster centres are treated as "visual words" and make up a codebook. At the retrieval stage, each image's local descriptors are extracted and clustered on those visual words, building the frequency histogram of each visual word. The global feature vector of each image is also composed of the frequency of each visual word. With these visual words frequency-based feature vectors, cosine similarity between the query image and each candidate image from the database is calculated to get retrieval results. Although the idea of BoV is essentially just an extension or transfer application from text-processing to image-processing, it influences many other methods employing local feature aggregation.

For instance, instead of simply using the frequency of each visual word to construct the global feature vector, the vector of locally aggregated descriptors (VLAD) [57] accumulates and concatenates residuals between each image local descriptor and the visual word to build the final compact image global descriptor.

The Fisher Vector [104] is also a successful feature aggregation method that can generate compact feature vectors for image representation. Given a set of pre-extracted image local features, the Fisher Vector transforms them into a fixed-size vector by the gradient of the log-likelihood function with respect to a set of parameter vectors. In [92], a Gaussian Mixture Model (GMM) is employed to aggregate the normalized gradient vectors of all local descriptors into a uniform Fisher Vector using an average pooling scheme. In a

[2]The name "bag of visual word" is also referred to as "bag of feature" or just "bag of word" in some other papers. Here to distinguish the origin of this idea in text-processing, we call this framework "bag of visual word (BoV)" throughout the thesis.

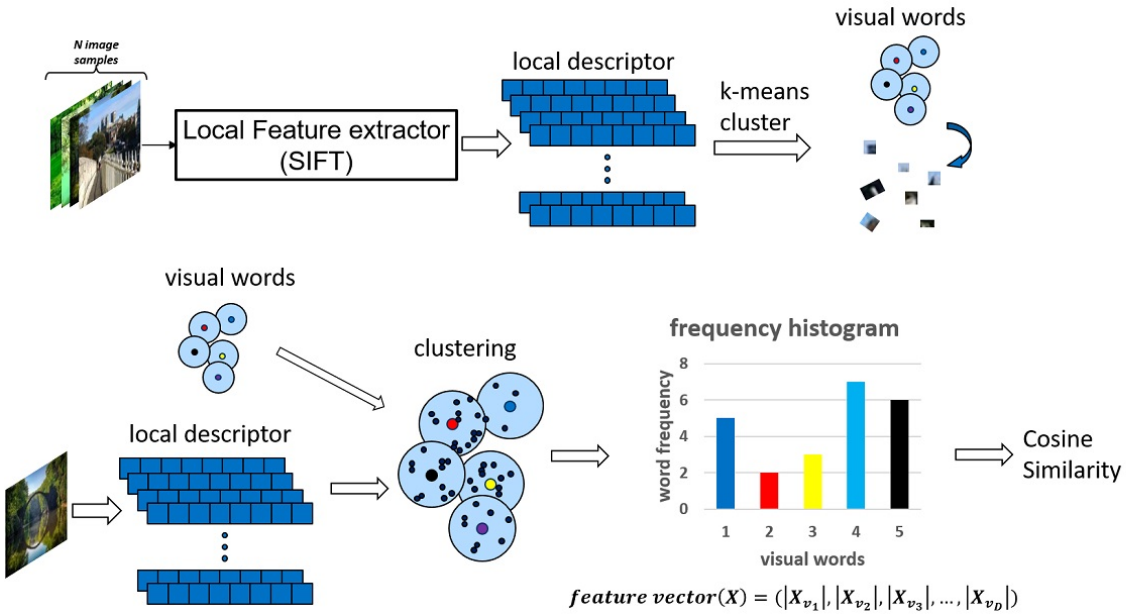


Figure 2.2: Illustration of bag of visual word pipeline [112]

way, the Fisher Vector can be regarded as a generalized representation of the BoV or a probabilistic version of VLAD.

Moreover, unlike previous works that attempt to aggregate local features into one compact global descriptor, the Aggregated Selective Match Kernel (ASMK) [121] encompasses many-to-many matching techniques with pre-extracted local feature vectors. Similar to the VLAD method, ASMK also first calculates the residual vector between each local feature vector and the corresponding visual word. However, instead of concatenating them into a compact global feature, ASMK aggregates the residual vectors corresponding to the same visual word by summation, resulting in a set of aggregated local feature vectors as the final representation of the original image. Then, a matching kernel is employed with these local features to perform the many-to-many similarity evaluation between images and get retrieval results.

2.2 Deep learning feature based methods

In this section, deep learning based feature extraction methods for CBIR are reviewed. It starts with an introduction to the development of convolution neural network (CNN) during the decade. Although CNNs had initially been used in other computer vision tasks, such as classification or object detection, they are also commonly applied as backbone net-

works, extracting feature tensors from RGB images for deep learning based CBIR. After that, we present more feature processing methods, aiming to transform the convolution feature tensor output by CNN into a more compact feature code for image retrieval.

2.2.1 Deep convolution neural network

The earliest multi-layer convolution neural network could be the ConvNet [68] back in 1989. ConvNet performs well for handwriting digit recognition and lays a foundation for modern 2-dimension CNN. Later in 1998, a well-known improved version of ConvNet, called LeNet [70, 69], was proposed. The success of LeNet advances the application of neural networks for recognition tasks in specific image domains, such as optical characters or fingerprints. Nevertheless, due to the limited computational power and access to the large-scale dataset in diverse image categories, CNNs did not perform that well in natural image recognition tasks at that development stage.

The first deep convolution neural network (DCNN) that brings milestone breakthroughs for general natural image recognition is the AlexNet [64]. AlexNet consists of 5 convolution layers followed by three fully connected layers. It is trained on ImageNet dataset [103], which contains more than 1 million images in 1000 classes, and achieves significant performance improvement over other conventional methods in the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)[3]. Since then, the DCNN has drawn great research interest in a long term and several attempts have been made to improve its performance in computer vision tasks further.

The VGG network [111] explored using deeper layers but small 3×3 convolution kernel to compose the CNN. This design brings several advantages. First, the deeper stack of convolution layers (with non-linear activation function, like ReLU [64], injected in between) makes the feature extraction function that the CNN implicitly learns at the training stage more discriminative. Second, compared with larger convolution kernel size, stacks of small convolution kernels could lead to the same receptive field size at their output while requiring fewer parameters. VGG gives good results on both image classification and localization tasks, ranking second in ImageNet Large Scale Visual Recognition

[3]https://image-net.org/challenges/LSVRC/2012/index

Challenge 2014 (ILSVRC2014)[4] competition.

Compared with VGG, which mainly works on the depth of network structure, GoogleNet [119], the champion of the ILSVRC2014 competition, proposes a deeper and wider CNN structure for image feature extraction. Instead of simply using single convolution layers, GoogleNet repeatedly utilizes the Inception module [119] towards the output layers to build the CNN. Each Inception module consists of 4 branches, where each branch contains convolution layers with different kernel sizes. In order to reduce the computation burden, before each expansive large kernel convolution layer, a 1×1 small kernel convolution layer is applied for dimension reduction. The design of the inception module enables each layer to process visual information at different scales, which is more intuitive and leads to better feature extraction. More importantly, it improves computational efficiency, making it feasible to build deeper architecture, which is beneficial for high-level feature learning [20] and mitigating the semantic gap [72].

Although many deep learning based computer vision works [37, 44, 36] have demonstrated to benefit from deep CNN models, a too deep architecture also makes the training harder. One problem is the vanishing/exploding gradient [11, 38]. Fortunately, this problem has been largely addressed by network parameter initialization [71, 105, 43] and intermediate layer normalization [50]. Another problem is the degradation: as reported in [42, 115, 45], excessively increasing the depth of CNNs by simply stacking convolution layers would result in saturated performance. To solve this problem, ResNet [45] introduces a deep residual learning framework into the CNN pipeline. By applying a short connection within each convolution bottleneck [45], the CNN model would learn a residual mapping instead of the original, unreferenced mapping [45]. Compared with a traditional plain network, the residual network is easier to be optimized and can continually benefit from deeper convolution layers. Specifically, ResNet successfully increases the depth of CNN from 16 (VGG16 [111]) or 22 (GoogleNet [119]) to more than 100 layers (ResNet-101 and ResNet-152) with significant improvements in the results. After that, more efforts are made to improve CNNs when used in computer vision tasks. For instance, the Inception-ResNet [118] employs the idea of residual connection into Inception network. The Xception [23] proposes a depth-wise separable convolution layer to replace the original convolution operation and reduce the computation cost while achieving a similar performance output.

[4]https://image-net.org/challenges/LSVRC/2014/

ResNeXt [140] proposes a more modularized convolution block, which also contains multiple branches but does not require specific designs for each branch as the inception module. DenseNet [47] extents the idea of the short connection between only specific layers to every other layer. Despite so many derivative CNN structures, ResNet still serves as the most commonly used backbone structure in recent deep learning based CBIR works [98, 87, 14, 146].

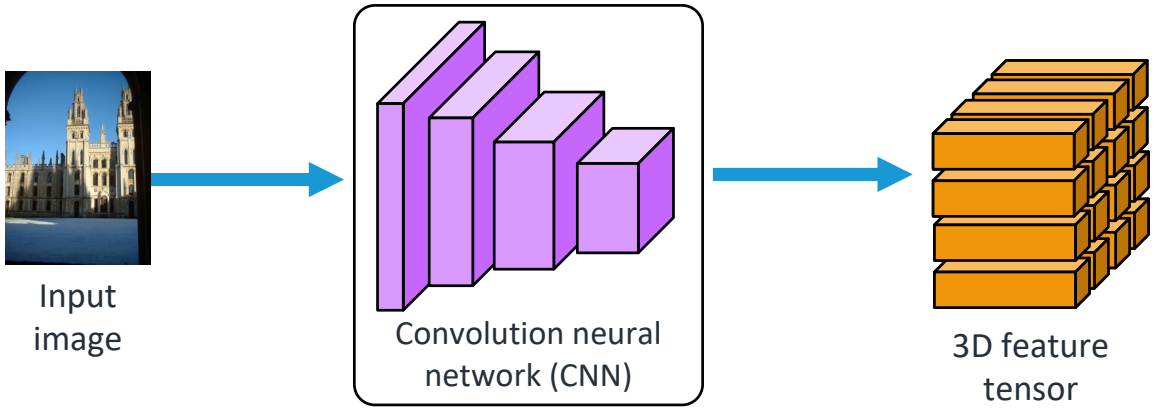


Figure 2.3: Illustration of image feature tensor extraction with CNN

Using CNN for image feature extraction helps people eliminate the reliance on domain knowledge for specialized algorithm design with low-level features. Instead, as shown in Figure 2.3, the hierarchical structure of CNN could automatically transform the input image into 3D feature tensors, where each entry on the feature tensor could be treated as a local descriptor, which contains rich local feature information and corresponds to a specific region from the original image. How to utilize this feature tensor for image retrieval is the central purpose of recent deep learning based CBIR research. In the following, this review will focus on representative methods that utilize the feature extracted by the convolution layers for the CBIR task.

2.2.2 Fully connected layer

The earliest work that utilizes the feature tensor output by convolution neural network (CNN) for content-based image retrieval is the Neural Code model [8]. In this work, AlextNet [64] is pre-trained on ImageNet [103] database with classification task then fine-tuned on the Landmark dataset [8] also with classification task. After fine-tuning, a fully connected layer is applied to map the 3D feature tensor from the last convolution

layer to a compact global feature vector to be used for image retrieval. In addition, the performance of the compressed Neural Code feature vector is also investigated. More specifically, principle component analysis (PCA) is used for feature dimension reduction leading to compressions to 256 or even 128 dimensions with almost no quality loss. In a way, it also proves that, without compression, the original feature output by CNN contains redundant information for image retrieval. After that, some other works try to further improve fully connected layer based features from different aspects. For example, the CBIR framework proposed in [59] concatenates features from multiple fully connected layers to build the final image global descriptor. The work from [114] achieves coarse-to-fine improvement by connecting different fully connected layers.

Different from the Neural Code model that utilizes a fully connected layer to transform the convolution feature tensor into a one-row feature vector in real-value elements, some other works explore using stacks of fully connected layers to approximate the real-value feature vector into binary hash codes for efficient and faster image retrieval [66, 75, 17, 58, 21], especially when it comes to large-scale databases. For instance, the approach from [74] fine-tunes a pre-trained CNN with additional latent layers on the target image domain to generate binary-like codes for coarse-level image retrieval.

2.2.3 Spatial pooling

Despite the success of fully connected layer based methods, the study from [99] demonstrates that spatial pooling is more appropriate for object retrieval than using fully connected layers. Thus, a series of spatial pooling methods are proposed to extract (or construct) a compact global feature vector from the output of the last convolution layer, including sum-pooling [7], max-pooling [124] and generalized mean pooling [98].

Considering an input image $\mathbf{I}$, after feeding forward a fully convolutional backbone network, it is mapped to a feature tensor $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$, where H, W, D represent the height, width and number of feature channels. The most intuitive pooling strategy to transform the 3D feature tensor $\mathbf{X}$ into a compact one-row feature vector could be the sum-pooling [7] and the max-pooling [124, 99]. They are also the earliest spatial pooling methods proposed for CBIR.

With $L = H \times W$, $l = 1, \ldots, L$ indicating entries on $\mathbf{X}$ and $\mathbf{x}_l \in \mathbb{R}^{1\times D}$ indicates the local feature vector from location l of $\mathbf{X}$. The simple sum-pooling[5] could be defined by:

$$\mathbf{V}_{sum} = \left(\frac{1}{L} \sum_{l}^{L} \mathbf{x}_l \right), \tag{2.1}$$

Let $\mathbf{X} = \{\mathbf{X}_d \in \mathbb{R}^{H\times W} | d = 1, 2, 3, ..., D\}$, $\mathbf{X}_d$ indicates the feature map slice at channel d from feature tensor $\mathbf{X}$, $x_{d,l}$ indicates the value at location l, The max-pooling could be defined by:

$$\mathbf{V}_{max} = [f_{1,L}, f_{2,L}, ..., f_{d,L}, ..., f_{D,L}], \text{with } f_{d,L} = \max_{l\in L} x_{d,l} \tag{2.2}$$

According to equations (2.1) and (2.2), it can be observed that the main difference between these two pooling methods is the way how the local context or neighbouring information is used. Sum-pooling uniformly extracts feature information from all locations of the convolution feature tensor, while the max-pooling only focuses on locations with the highest activation value across all channels. Although it is argued that sum-pooling performs better than max-pooling when the image representation is PCA-whitened [7], the work of [124] proves this is not always true in the context of object localization or when it comes to describing region level features.

After them, a new spatial pooling method called generalized mean pooling (GeM) was proposed, which could be defined by:

$$\mathbf{V}_{GeM} = \left(\frac{1}{L} \sum_{l}^{L} {\mathbf{x}_l}^p \right)^{\frac{1}{p}} \tag{2.3}$$

Where p is a power coefficient. By setting a proper value for p, the GeM pooling leads to the best retrieval performance compared to other spatial pooling methods.

As the similarity measure between these pooled global features is normally performed with cosine similarity, at the training stage, the optimization of these CNN-based spatial pooling CBIR models is also normally conducted with loss functions that optimize the cosine similarity (or L2 distance) between labelled image pairs, such as contrastive loss [24] or triplet loss [4]. Recent works have proposed a more comprehensive loss function

[5] In some places, it is also referred to as global average pooling

for CBIR model training. For example, the listwise loss [101], ranked list loss [133] and Smooth-AP loss [13] all propose a training framework to directly optimize the average precision or rank order within each image batch, which is more intuitive and leads to better retrieval accuracy. Moreover, unlike the learning procedure mentioned above that only optimizes the metric distance between specific image samples at each training step, the NCA-proxy loss [82] proposes to represent image class with a trainable proxy feature and directly optimize the distance between the training sample image and each class proxy at every training step. This proxy-based loss leads to higher retrieval accuracy and speeds up model convergence during the training. The only drawback of this proxy-based learning could be that it requires additional class label information for each training image.

2.2.4 Convolution feature aggregation

Unlike methods mentioned above that tend to extract the compact global feature vectors by one single forward through the network structure, some other works treat each entry of the feature tensor output by the final convolution layer as a dense local feature vector and combine these convolution local features with classic feature aggregation methods.

For instance, the Bag of Local Convolutional Features (BLCF) [81] represents a method combining CNN-based local features with the bag of words (BoV) [112]. BLCF adapts the pipeline of the bag of visual words (as shown in Figure 2.2) to build a compact feature vector selected from the convolution feature tensor. Basically, BLCF replaces the SIFT local features from Figure 2.2 with local features output by pre-trained CNN. Similarly, in [149], VLAD [57] is adapted and combined with CNN-based local features. Moreover, NetVLAD [4] modifies VLAD as an end-to-end trainable layer at the tail of the CNN structure. The experimental results show that the trainable VLAD outperforms the local feature fusion methods, which are not based on deep learning.

2.2.5 Self-supervised feature learning

Apart from those works mentioned above that require annotated image data for supervised feature representation learning, other works try to train the whole model in a self-supervised (unsupervised) manner. At the early stage, self-supervised feature learning

for image retrieval is mainly based on hashing. It could either utilize some generative mechanisms [27, 113, 158, 31] or apply with some graph-based techniques [107, 108]. The recent work the Self-supervised Product Quantization (SPQ) network [54] adapts the self-supervised contrastive learning framework [18, 67, 19] to learning visual representation for content-based image retrieval. Instead of directly optimising features between annotated image pairs, SPQ applies visual transforms on a single input image, resulting in descriptors corresponding to two different "views" of the same image. Then, the cross-similarity between the correlated deep descriptor and the product quantized descriptor is maximized and the whole model could be optimized in an end-to-end manner.

2.3 Attention mechanism for CBIR

In the following, CBIR papers with different types of attention mechanisms are reviewed. Generally, depending on how the attention weights are applied to the image features, the attention mechanism could be divided into two groups: spatial attention and channel attention.

2.3.1 Spatial attention

Spatial attention focuses on where is the important part of the current vision task. It is usually used for weighting the extracted image features according to the location. According to the origination of the attention information, spatial attention could be either guided by human perception or driven by training data.

Human perception guided attention

Spatial attention has been applied for feature selection and weighting for more than ten years [41, 52]. Unlike later trainable neural network based attention modules that can automatically learn the attention mechanism in an end-to-end manner from large-scale training data, conventional methods can only obtain visual attention through a hand-crafted attention mechanism or algorithm. Under this circumstance, one popular visual

attention mechanism was saliency. Saliency refers to the local region highlighting or re-weighting mechanism that matches with human perception of the input image. In a way, it reflects the probability that each region is likely to catch human attention. Several algorithms have been used to model human perception based saliency from images such as the classic algorithm of Itti & Koch [53], Graph-based Visual Saliency (GBVS) [41], ittiKoch [52], SUN [152] and FTS [1]. These saliency detection methods are normally embedded into the feature extraction procedure for local feature selection or re-weighting, leading to more comprehensive feature output and more reliable model performance.

For instance, based on the classic saliency method Itti & Koch [53], the global feature extraction model: "gist" [109] proposes a framework that efficiently generates both the saliency map and global feature of the input image from shared low-level feature information, including orientation, colour and intensity. The gist feature has been applied in several image search works [65, 56, 125].

The work from [89] gives a comprehensive study of combining saliency map and low-level features for content-based image retrieval. In this work, each image is segmented into sub-regions, and each region feature is extracted based on colour, texture and contrast. The saliency map is extracted with a separate saliency module. Figure 2.4 shows the saliency map provided for a given image by various saliency extraction methods. The saliency map is used to select saliency regions and give these regions a higher weight when applying the similarity measure. With these region-level features, Earth Mover's distance [102] was utilized for similarity measure at the retrieval stage.

Figure 2.4: Saliency maps generated by different saliency methods. Image taken from [89].

After stepping into the age of deep learning, saliency was also applied in the CNN-enabled feature extraction pipeline for image retrieval. For instance, the work from [80] proposes a CBIR framework with two branches, one for feature extraction and one for saliency extraction. The feature extraction branch is similar to the Bag of Local Convolutional Features (BLCF) [81]. For the saliency extraction, the author tested different methods, including some deep network based saliency extraction methods such as Saliency GAN [88], or the conventional saliency method like GBVS [41]. The visual word assignment map [80] and the saliency map are fused by element-wise multiplication. In other words, the saliency map serves as a weight mask on the assignment map, and then a weighted frequency histogram of visual word [112] is built to get the feature vector. The work from [80] is not an end-to-end trainable deep learning model for CBIR. It just combines a pre-trained deep learning based feature extractor and a pre-defined saliency extraction method. In other words, the feature extraction branch and the saliency extraction branch can not be jointly fine-tuned or optimized on the target image domain.

Another model, the Two-Stream model [145], also uses one branch for feature map extraction and another for saliency information extraction. The output of the saliency map branch and feature extraction branch are fused and fed into fully connected layers. The output of the fully connected layer serves as a global feature vector for image retrieval. Compared with the work from [80], the Two-stream model also utilizes pre-trained feature extractors and saliency extractors but makes them trainable on the target image dataset.

However, as the human-perception-based saliency module is designed to model how humans would pay attention to locations over the image, regions that are likely to catch human attention may not always be consistent with the actual region (object) of interest. This is likely to happen when the target object is not salient or more eye-catching distractors are nearby in the input image. As a result, the saliency attention module may look toward regions outside the correct target.

Training data guided attention

Instead of imitating how humans would pay attention to the image content, another approach to spatial attention mechanisms is driven by the training data. This kind of attention is usually generated by a trainable network branch which is end-to-end optimised

with the whole network model at the training stage or would directly derive its feature output by an intermediate layer.

For example, in [39, 40], a Region Proposal Network (RPN) [100] is implemented to pick out regions of interest, improving the max-pooling global feature vector for image retrieval. The whole model is end-to-end trainable and gives better retrieval results than simple global pooling. The Weighted Generalized Mean pooling (WGeM) [138] applies a trainable spatial weighting module by adding an extra convolutional layer at the end of a CNN backbone structure. It can effectively localise objects of interest while ignoring redundant regions. However, the spatial weighting may fail when the target object is not discriminating or not matching the training data [138]. The Deep Orthogonal Local and Global (DOLG) [146] also utilises a convolution layer based spatial attention module for local feature re-weighting from a shallower convolution layer. Then, a comprehensive global feature extraction pipeline, in which an Orthogonal Fusion module is implemented to complement the global feature vector with re-weighted local features, leading to the current state-of-the-art results for CBIR.

The Second-Order Loss and Attention for image Retrieval (SOLAR) [85] explored the co-relations between each location from the CNN feature map using the second-order spatial information. Unlike the attention methods mentioned above that would only generate one attention map applied on the CNN feature map, in the SOLAR pipeline, for each location, a second-order attention map is generated to indicate its connection to all other locations. SOLAR is trained on the Google Landmark Dataset (GLD) [87], so the model tends to treat all landmark relevant regions as regions of interest. For the irrelevant locations that correspond to common background noises, such as grass or sky, the second-order attention will sparsely distribute over all landmark-like regions. Meanwhile, for a location from a landmark, the second-order attention would highlight the most distinctive part of that landmark.

Unlike the methods mentioned above that introduce extra trainable layers or parameters for spatial attention generation, the architecture proposed by Tolias *et al.* for HOW [123] directly uses the L2-norm of each entry in the convolution feature tensor as spatial attention for local feature selection. The chosen local features are applied with the Aggregated Selective Match Kernel (ASMK) [121] for many-to-many local feature matching. Within the framework of HOW, k-means clustering is also applied with local features to build

a codebook, which is similar to the bag of visual words [112]. The HOW-MDA [137] develops the Multiple Dynamic Attention (MDA) module for refining the local features resulting in better retrieval results than HOW.

Compared with human-perception-based saliency attention, the spatial attention driven by training data is specifically optimised on the target image domain during the training. At the evaluation stage, it tends to highlight training data relevant regions (objects). Most recent works also consider attention mechanism driven by training data [14, 146, 123]. However, as mentioned in Chapter 1, most existing attention mechanisms are query non-sensitive. Therefore, they may still fail when similar class distractors surround the target object, or there are multiple potential objects of interest in the input image.

2.3.2 Channel-attention

As each convolution kernel could be treated as a feature detector sensitive to specific patterns [151], channel attention aims to find which channel of the feature tensor output by the convolution layer contains the most critical information for the current task.

Motivated by the finding that the sparsity pattern of convolution feature channels could contain discriminative information and those infrequently activated feature channels could still carry important signals for image retrieval, the cross-dimensional weighting and pooling (CroW) model [60] proposes a non-parametric channel-wise attention mechanism to boosts the contribution of rarely found, but important features before the global pooling operation. Furthermore, the fully cross-dimensional weighting pooling (FCroW) [129] further improves CroW by incorporating multi-layer fusion and more comprehensive weighting into the feature extraction pipeline.

The Multiple Saliency and Channel Sensitivity Network (MSCNet) [139] utilizes the Gram matrix [35] for correlation analysis between feature channels. It is combined with the sparsity-sensitive channel weights (SSW) [139, 60] to construct the channel-wise attention module, leading to more discriminative feature extraction.

The part-based weighting aggregation (PWA) [143] considers that each convolution feature channel implicitly contains a different semantic meaning and represents a specific part of

the input image. Based on the variance of each feature channel output, a "probabilistic proposals" [143] is built to emphasize those channels corresponding to discriminative regions.

2.3.3 Co-attention

Co-attention has drawn research interest from various computer vision tasks but was hardly considered for CBIR.

For instance, the query-guided end-to-end person search network (QEEPS) [83] proposes three novel query-guided sub-networks: QSSE-Net, QRPN and QSimNet that would embed query information into the CNN feature channel by re-weighting, using relevant region proposal or considering the similarity score prediction, respectively.

The co-attention and co-excitation (CoAE) framework [46] utilizes the non-local operation [132] to explore the correlated evidence revealed by the query-target pairs. The extended feature maps are then channel-wise re-weighted by using the squeeze-and-co-excitation (SCE) technique. The Region Proposal Network (RPN) [100] selects relevant regions based on the extended target image feature map. RPN can predict relevant regions with respect to the query content even when images from the query class have not been seen during training.

The SiamMask [130] uses depth-wise cross-correlation to generate response maps of the target image with respect to the query. Then the response map is fed into the convolution layers for pixel-wise classification to generate binary co-attention masks.

2.4 Re-ranking

Among recent CBIR works, there are three frequently used re-ranking strategies for initial retrieval result refinement: spatial verification [93, 106, 6], query expansion [26, 25, 122] and diffusion [51].

Spatial verification utilizes local features to perform spatial information matching for re-

trieval result re-ranking. It is normally achieved with the help of the RANSAC algorithm [33]. The DEep Local Feature (DELF) [87] could be a representative two-stage local feature model that utilizes local features to perform spatial verification for initial retrieval results re-ranking. It implements a score function with two processing layers on top of the final convolution layer for relevant local feature selection. During the first initial retrieval stage, the compact global feature vector is built by a weighted sum of selected local features. During the re-ranking stage, after dimension reduction, geometry verification is performed with these local features to get the final retrieval result. Based on DELF, Detect-to-Retrieve (D2R) [120] proposes the Regional Aggregated Selective Match Kernel (R-ASMK), which unifies the region of interest detection, regional local feature aggregation and the similarity measure into one pipeline. Deep Local and Global features (DELG) model [14], also based on DELF, unifies the training procedures of global and local features into a single pipeline and further improves the performance of this two-stage image retrieval framework.

Query expansion (QE) was originally proposed as a standard method for performance improvement of text processing and retrieval. The core idea of QE consists of reusing high-ranked initially retrieved items to construct a new query, providing more comprehensive query information and getting better retrieval results in the second round of search. In [26], the standard average query expansion (AQE) was first introduced to the field of CBIR and it was widely applied in compact global feature based CBIR works [7, 60, 124]. The α-weighted query expansion (αQE) [98] improves the AQE by introducing a cosine similarity measure to each top-ranked retrieved image, leading to a more robust query expansion strategy for the CBIR task. As query expansion is sensitive to the initial retrieval accuracy, some works propose to use spatial verification, reducing the negative impact caused by the false positive match among top-ranked retrieval results [25]. The Hamming Query Expansion (HQE) [122] proposes a simpler but still effective method for reliable image selection before performing query expansion, leading to better retrieval results even without spatial verification.

Diffusion [51] could be treated as an extension of query expansion, as QE only acts on top-n retrieval results for the new query item building, while diffusion performs an online exploration of the nearest neighbour, with respect to the query, by constructing a neighbor graph of the entire dataset. Like query expansion, diffusion is also sensitive to the

initial retrieval results. The deep spatial matching (DSM) [110] proposes a deep learning feature based spatial matching framework to refine the initial retrieval results before performing diffusion. This approach combines the advantage of both local and global features, achieving a larger performance improvement margin than directly employing diffusion with the initial retrieved images.

2.5 Benchmark datasets

Commonly used benchmark datasets for CBIR model performance evaluation are listed below:

INSTRE [131] is an instance-level retrieval dataset collected from multiple source and has been utilized in many computer vision tasks, including content-based image retrieval. It is composed by two subsets: INSTRE-S and INSTRE-M. The former one contains 23,070 images in 200 categories and each image only contains one object of interest. The latter one consists of 5,473 images and each image contains two instances from 100 object categories.

University of Kentucky Benchmark Dataset (UKB) [86] contains 10,200 images in 2,550 groups. Each group contains 4 images of the same object under different acquisition conditions. By default, the retrieval accuracy on UKB dataset is reported with the average number of same-object images within the top 4 results.

INRIA Holidays Dataset (Holiday) [55] contains 1,491 images collected from personal holiday photo albums. Some images are taken on purpose with different acqusition condition, such as rotation, illumination change, different view point, etc. All images are divided into 500 groups, with Each group corresponds to a different scene or object.

Oxford Building Dataset (Oxford5k) [93]: Oxford5k dataset contains 5062 images which are collected from Flickr with 17 tags, such as Balliol Oxford, Christ Church Oxford, Hertford Oxford, Jesus Oxford, Keble Oxford, etc. All images are manually annotated and it contains retrieval ground-truth images in 11 categories. For each landmark category, there are 5 query images, so it gives 55 images in all as queries for image retrieval evaluation.

Pairs Building Dataset (Paris6k) [94]: Paris6k dataset consists of 6412 images and is also collected from Flickr by searching for 12 different particular Paris landmarks, such as La Defense Paris, Eiffel Tower Paris, Hotel des Invalides Paris, Louvre Paris. And it also gives 55 queries with which the image retrieval model can be evaluated.

By considering an additional set of 100K distractor images collected from Flickr, Oxford5k and Paris6k can be expanded to Oxford105k and Paris106k, providing a more challenging image retrieval scenario for CBIR model performance evaluation.

Revisited Oxford (ROxf) and Paris (RPar) Datasets [96]: ROxf/RPar are expanded versions of Oxford [93] and Paris [94] datasets after removing the images with incorrect annotation and adding several new query images. ROxf contains 4993 images while RPar has 6322 images. Both datasets contain 70 query images. The ground-truth matching images to each query image are divided into 3 groups, *Easy, Medium, Hard*, according to the level of difficulty in assessing the similarity of their image representation with the corresponding query. In addition, R1M [96] is a new distractor set containing 1 million unbiased high-resolution (1024×768 pixels) images for ROxf and RPar.

Dataset	Method	Result
INSTRE	BLCF-SalGAN [80]	69.8
UKBench	R-MAC [40]	3.90
Holiday	R-MAC [40]	94.0
Oxford-5k	WGeM [138]	88.8
Oxford-105k	WGeM [138]	85.6
Paris-6k	R-MAC [40]	93.6
Paris-106k	DELF [87]	81.7
ROxf-5k (*hard*)	DOLG [146]	64.9
ROxf-5k+1M (*hard*)	DOLG [146]	51.6
RPar-6k (*hard*)	DOLG [146]	81.7
RPar-6k+1M (*hard*)	DOLG [146]	62.9

Table 2.1: Quantitative retrieval results on common benchmark datasets. All results are reported with mAP [93] except on the UKB dataset, which is reported with the average number of same-object images within the top 4 results. The "Method" shows the name of the model.

Table 2.1 presents the current state-of-the-art retrieval results over these common benchmark datasets. For the ROxf/RPar datasets, we focus on the most challenging *hard* set of them.

2.6 Summary

Generally speaking, CNN-enabled CBIR could be divided into two categories according to the feature representation level: global feature and local feature. Global feature methods tend to extract a compact feature vector for each image by single forward processing passing through the network. The feature tensor provided by the final convolution layer is used by a fully connected layer or by a global spatial pooling layer. Then, a similarity measure is directly performed with these global feature vectors corresponding to a query and a search image, usually by calculating the L2 distance between them after being normalized.

Local feature methods treat each entry of the feature tensor output by the last convolution layer as a feature representation of a specific region from the input image. Depending on the usage of local features, they could be further categorized into different approaches. The first category implements a separate aggregation method to encode local features into a compact feature vector. On the contrary, instead of aggregating local features into a compact code, the second category keeps several local features from each image and employs a similarity measure in a many-to-many manner. The final category of local feature methods uses the spatial information of each local feature from the original convolution feature tensor to perform spatial verification only for the initial retrieval results refinement at the re-ranking stage.

Most recent CBIR methods, such as DELF [87], DELG [14], DOLG [146], or HOW [123] are based on the spatial attention driven by training data. However, all these existing attention mechanisms are all query non-sensitive, they just predict the likely region of interest based on the knowledge learned during the training regardless of the actual query information. As discussed in Chapter 1, these query non-sensitive attention tend to fail when the object is not salient or surrounded by similar class distractors. On the contrary, introducing the query sensitive co-attention mechanism into the CNN-enabled CBIR pipeline represents a promising direction for research studies aiming to achieve further performance improvement in CBIR.

Chapter 3

Conditional attention network

3.1 Introduction

Existing spatial attention mechanisms for content-based image retrieval are all query non-sensitive: they only consider a single candidate image as input for spatial attention map generation. This kind of attention works great for filtering out training data irrelevant clutters or background noises. Moreover, once the training procedure is done and the module is fixed, the resulting attention map is also fixed for each candidate image. However, in real-life image retrieval situations, one candidate image could contain multiple training data relevant items or potential regions of interest. The actual region of interest will subjectively vary with the query content or the search purpose of the CBIR system user. When the target object is not salient or surrounded by distractors relevant to the training data, these query non-sensitive spatial attention modules are very likely to focus on incorrect regions and ignore the object of interest, as shown in Figure 1.3. To solve this problem, only the query sensitive attention: co-attention could be a good solution. Query sensitive means the attention module's output changes with the query content. In other words, it considers the content of both the query image and candidate image for spatial attention generation. The resulting co-attention map is supposed to be consistent with the input query content.

In this chapter, a co-attention model, namely the Conditional Attention Network (CANet), is proposed for localizing the object (region) of interest from the candidate image that

matches the content of the query image. Unlike previous query non-sensitive attention modules, which rely only on the single candidate image's feature tensor to generate attention maps, the CANet considers both candidate and query images as input. Their features are fused and transformed into a focused spatial attention map by stacks of convolution layers. The whole model is end-to-end trainable and can be combined with generic CNN-based feature extraction methods for feature re-weighting, boosting the original method's retrieval performance.

An essential training requirement of a deep learning model is access to a large enough annotated dataset. However, most image retrieval training datasets would only provide matching image pairs or different classes of images without ground-truth matching regions or target objects' locations. It is impractical to manually mark matching regions within large datasets. To obtain accurate region-level match labels for training the proposed conditional attention model, the pre-trained key-point detector SuperPoint [30] is considered to find correspondences of existing matching image pairs. Then a simple but effective trick is proposed to transform matching key-points into matching regions. The SuperPoint model is a fully self-supervised model. Accordingly, the training of the proposed conditional attention network does not require any extra ground-truth region-level labels.

The rest of this chapter contains the following sections. The design of the proposed CANet structure is presented in Section 3.2 while the pipeline of region-level training data generation is outlined in Section 3.3. Section 3.4 details how CANet can be integrated into a deep learning CBIR pipeline. Section 3.5 includes experiment setting detail, image retrieval results and comparison to existing CBIR works. Ablation studies and more discussion are provided in Section 3.6. The conclusions of this chapter is drawn in Section 3.7.

3.2 Conditional attention network structure

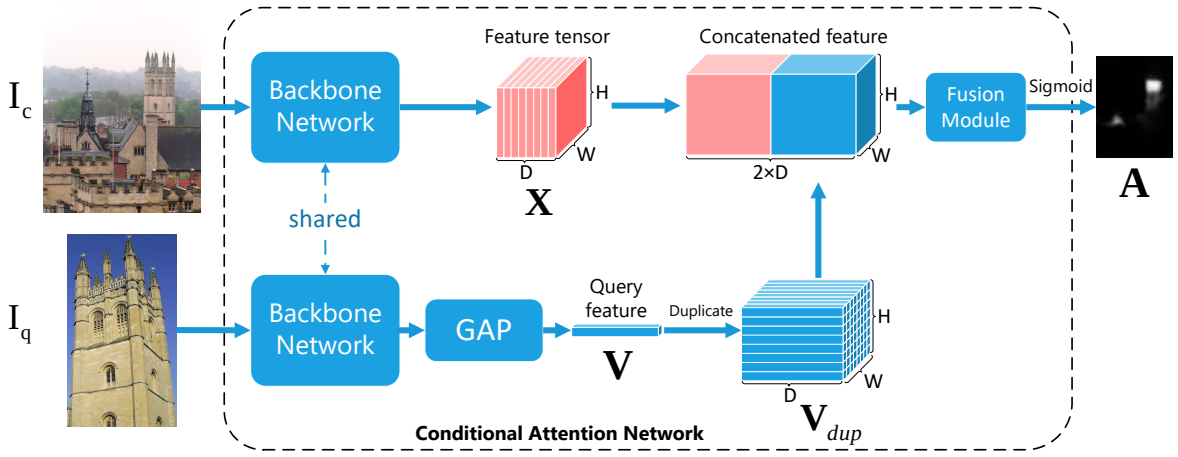
In the following, the characteristics of the Conditional Attention Network (CANet) architecture are described.

Network architecture. The proposed Conditional Attention Network (CANet) is designed to define Regions Of Interest (ROI) in candidate images under the condition of the content in the query image. Its architecture is shown in Figure 3.1. The co-attention map generation pipeline consists of three processing stages: visual encoding, feature fusion and attention map generation.

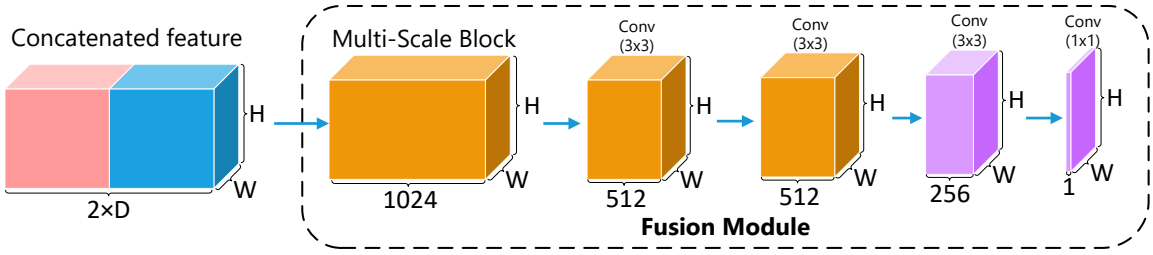
Visual feature encoding. A convolution neural network serves as the backbone network to encode the feature tensor from both query and candidate images. Given the candidate image $\mathbf{I}_c$, the output of the backbone network is a feature tensor $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$, where H, W, D represent the height, width and dimension (channel count) of the feature tensor. The query image $\mathbf{I}_q$ is also fed into the backbone network followed by Global Average Pooling (GAP), yielding the query global feature vector $\mathbf{V} \in \mathbb{R}^{1 \times D}$. The goal is to compare the query feature vector $\mathbf{V}$ with the information from each location of the candidate feature tensor $\mathbf{X}$.

Feature fusion. As shown in Figure 3.1 (a), the proposed attention model fuses the 3D feature tensor $\mathbf{X}$ of a candidate image $\mathbf{I}_c$ with the global query feature vector $\mathbf{V}$ representing $\mathbf{I}_q$. First, the feature tensor $\mathbf{X}$ is location-wise L2 normalized. The query feature vector $\mathbf{V}$ is also L2 normalized then expanded to $\mathbf{V}_{dup} \in \mathbb{R}^{H \times W \times D}$ by simple duplication. The feature tensors $\mathbf{X}$ and $\mathbf{V}_{dup}$ are concatenated and then fed into a fusion module for feature fusion and the final co-attention map generation.

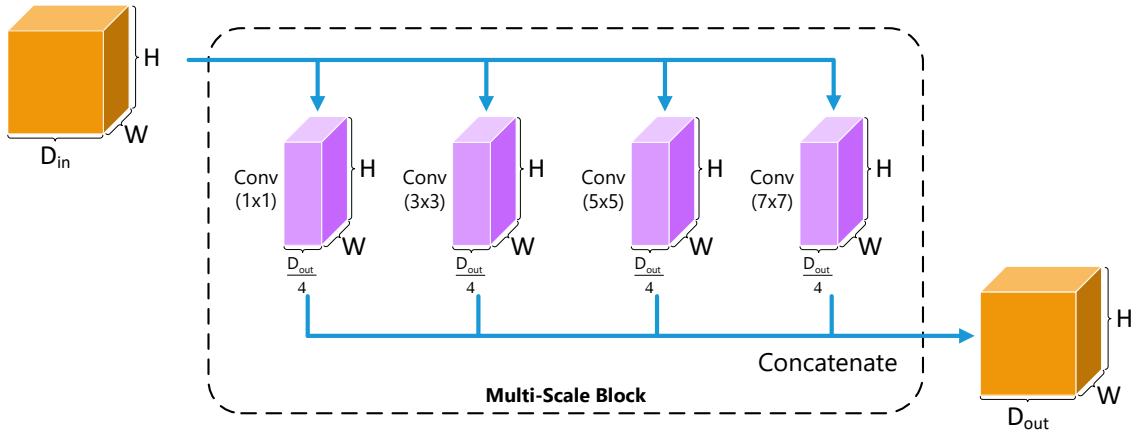
The pipeline of the fusion module is shown in Figure. 3.1 (b). Within the fusion module, the concatenated feature tensor is processed by several multi-scale convolution blocks followed by convolution layers for further channel reduction. The final convolution layer with convolution kernel size of 1×1 serves as the output head to transform multiple channel feature tensors into a single channel attention map. The major component of this fusion module is the multi-scale convolution block. The details of multi-scale convolution block are shown in Figure 3.1 (c). Given that the target object size can vary, due to changes in the image acquisition conditions, with different candidate images, each multi-scale convolution block consists of several convolution layers with different kernel sizes: $\{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$. Each convolution layer takes a feature tensor with D_{in} channel as input, and outputs a feature tensor with $\mathrm{D}_{out}/4$ channels. All convolution layers' outputs are then concatenated, resulting in the final output feature tensor with D_{out} channels. Intuitively speaking, the fusion module ensures the interaction and evaluation



(a) Illustration of the Conditional Attention Network global structure.



(b) Illustration of the feature fusion module.



(c) Illustration of the multi-scale convolution block.

Figure 3.1: The architecture of the proposed Conditional Attention Network (CANet).

of the consistency between the candidate image's local feature from each location and the global query feature in a trainable manner. In addition, the multiple kernel size design of the multi-scale block enables each location of the resulting feature tensor with different receptive field sizes, being aware of more context information.

Co-attention generation. After the fusion step, a Sigmoid activation function is employed to normalize each location value within range of $[0, 1]$ and generate the final one-channel co-attention map $\mathbf{A} \in \mathbb{R}^{H \times W \times 1}$ for the candidate image $\mathbf{I}_c$ under the condition of the query image content from $\mathbf{I}_q$. In a way, the generated co-attention map models the likelihood that each location from $\mathbf{I}_c$ would match with $\mathbf{I}_q$.

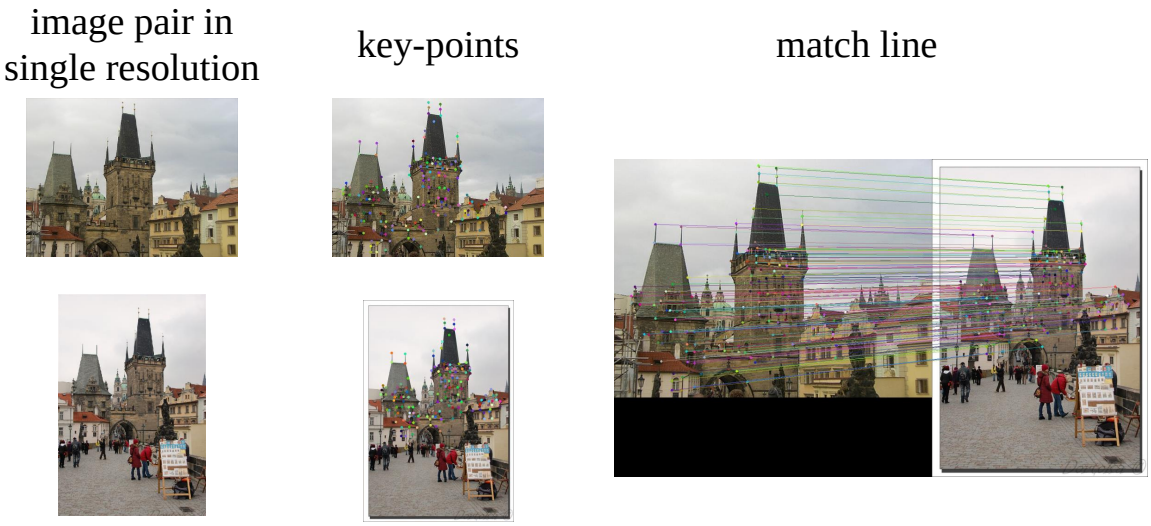
3.3 Data generation and training

In this section, the pipeline of training data generation and the training procedure are illustrated.

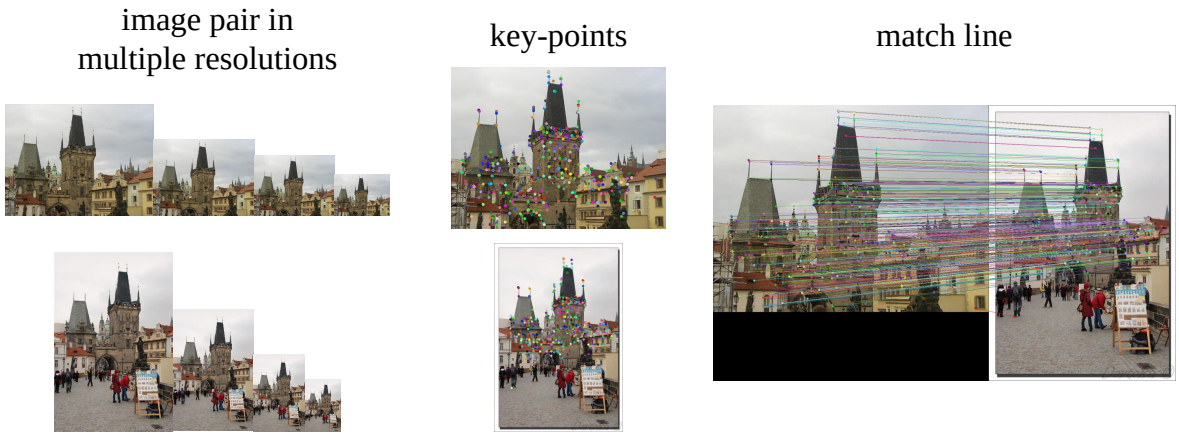
3.3.1 Defining image correspondence features

In the following, it is assumed that pairs or sequences of corresponding images have been collected, which represent sections of the same scene but are acquired at different times, under different conditions, and characterized by different image acquisition parameters in general. These paired images, displaying parts of the same scene, are used to find the matching regions and corresponding ground-truth attention maps, which serve as training data for the proposed CANet.

First, the SuperPoint model [30] is used as an intermediate feature descriptor to extract local image descriptors and find key-point correspondences among matching image pairs, as shown in Figure 3.2 (a). In order to obtain more robust key-points, for each matching image pair, both query and positively matching images are separately resized while keeping their original image aspect ratio. The key-point detection and matching are performed with multiple resolutions for each image. In practice, 4 different resolutions with $\{128, 256, 362, 512\}$ for the long side are considered. Figure 3.2 (b) shows how the



(a) Visualization of SuperPoint key-point detection with single image resolution (image long side:{362}) as input.



(b) Visualization of SuperPoint key-point detection with multiple image resolutions (image long side:{128, 256, 362, 512}) as input.

Figure 3.2: Visualization of SuperPoint key-point detection with different image resolutions.

matching key-point pairs detected by SuperPoint with different input image resolution works. All detected key-point coordinates are scaled and projected back into the original image for visualization. From the examples in Figure 3.2, it can be observed that when considering multiple input image resolutions, the resulting key-points are more dense and comprehensive. As a result, even matches from some tiny detail regions can be accurately detected.

Then, for each image, the key-point map $\mathbf{M}$ that represents the locations of the non-parametric distributions defined by the density of matching key-points is created. The key-point map is much smaller than the original image while keeping the original image's

ratio. As shown in Figure 3.3, projecting key-points detected by the SuperPoint model from the original image to a smaller key-point map $\mathbf{M}$ could be treated as splitting the original image with small grids and evaluating the density of the key-points within each grid location, resulting in a localized clustering. In addition, among the key-point regions, like the main tower structure in Figure 3.3 that contain dense matching key-points, after projection, neighbouring key-points would be connected with each other into larger and more reliable structures on the key-point map.

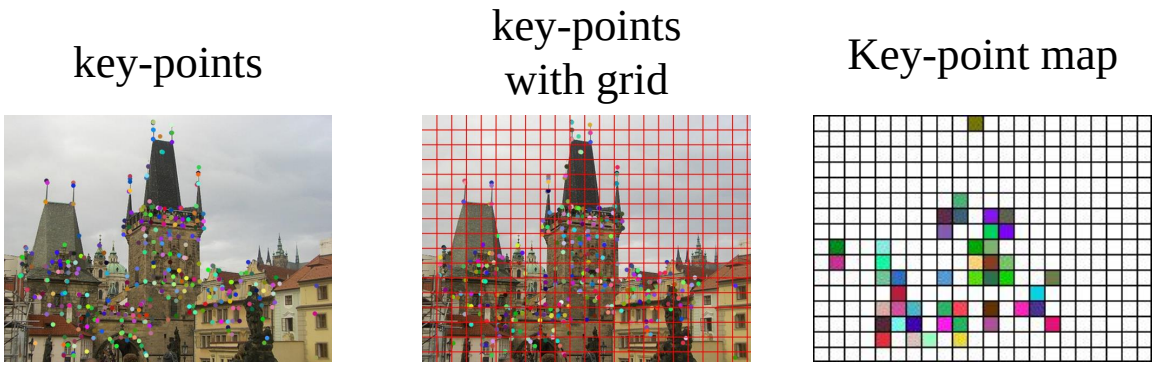


Figure 3.3: Merging and concatenating matching key-point into local regions on a grid leading to well defined structured regions.

As shown in Figure 3.4, matching key-points of the query image and the corresponding positive image are separately projected to the key-point map $\mathbf{M}_Q$ of size $H_{MQ} \times W_{MQ}$ and $\mathbf{M}_P$ of size $H_{MP} \times W_{MP}$ while keeping the original aspect ratio of the image. With key-point maps, two criteria are considered for defining the matching regions: 1. The region is defined within the top-left and bottom-right key-points; 2. The region is defined by connected key-point regions from the key-point map, which is larger than a small neighbourhood such as that of 3×3 pixels. All locations within matching regions are labelled by 1 or 0 otherwise, resulting in the final binary ground-truth attention map. Based on the SuperPoint outputs, each image pair can generate several estimates of matching regions $\hat{\mathbf{A}}$. In other words, one positively paired image can generate several sets of ($\mathbf{I}_q$, $\mathbf{I}_c$, $\hat{\mathbf{A}}$), which can be used as training data.

Choosing the right size for the key-point map is important. If the key-point map is too small, the precision of the generated matching region will be very low. If the key-point map size is too large, then the key-points will be too sparse to localize and represent the appropriate regions in the given image pair. According to the experimental tests, setting the long side of key-point map around 20 pixels can generate accurate matching

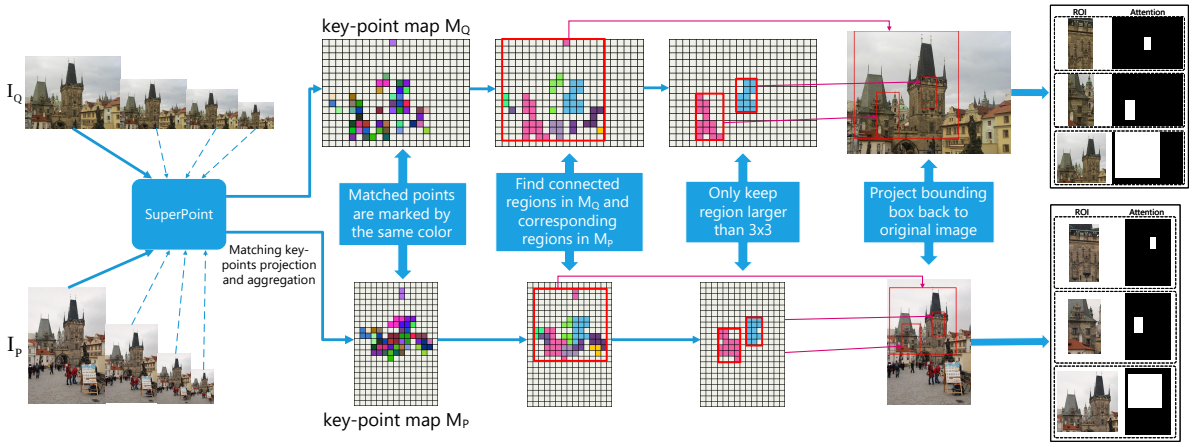


Figure 3.4: The pipeline of training data generation. Selected matching regions are projected back into the original images in order to define the regions of interest. The long side of all key-point maps and the final generated ground-truth attention map is considered as 22 in the experiments while preserving the original image ratio.

region pairs and corresponding ground-truth co-attention maps for later CANet training. Examples of generated training data are shown in Figure 3.5. Some generated matching regions do not even correspond to an intact building or object, but they are just some image regions characterized by certain properties. However, according to the experimental results, even when CANet is trained with such matching local patches, it can generate good co-attention maps.

3.3.2 Learning image correspondences

A good example of image dataset containing a sizeable number of annotated matching image pairs is the image tuple dataset rSfM-120k [97, 98], which contains 91,642 images divided into 551 clusters, while 181,697 matching image pairs are annotated. Each pair of matching images can be used for generating matching key-points and correspondence regions as described in Section 3.3.1. Each annotated image pair could generate 1 to 4 sets of local matching regions along with corresponding ground-truth co-attention maps. Meanwhile, those matching image pairs that are not able to generate any local matching regions will not be used for training.

During the training, each input is a data tuple consisting of 1 query image, 1 positive image and 5 negative images, which are randomly selected from the unmatching images. Within each tuple, given the positive image pair, several pairs of query regions

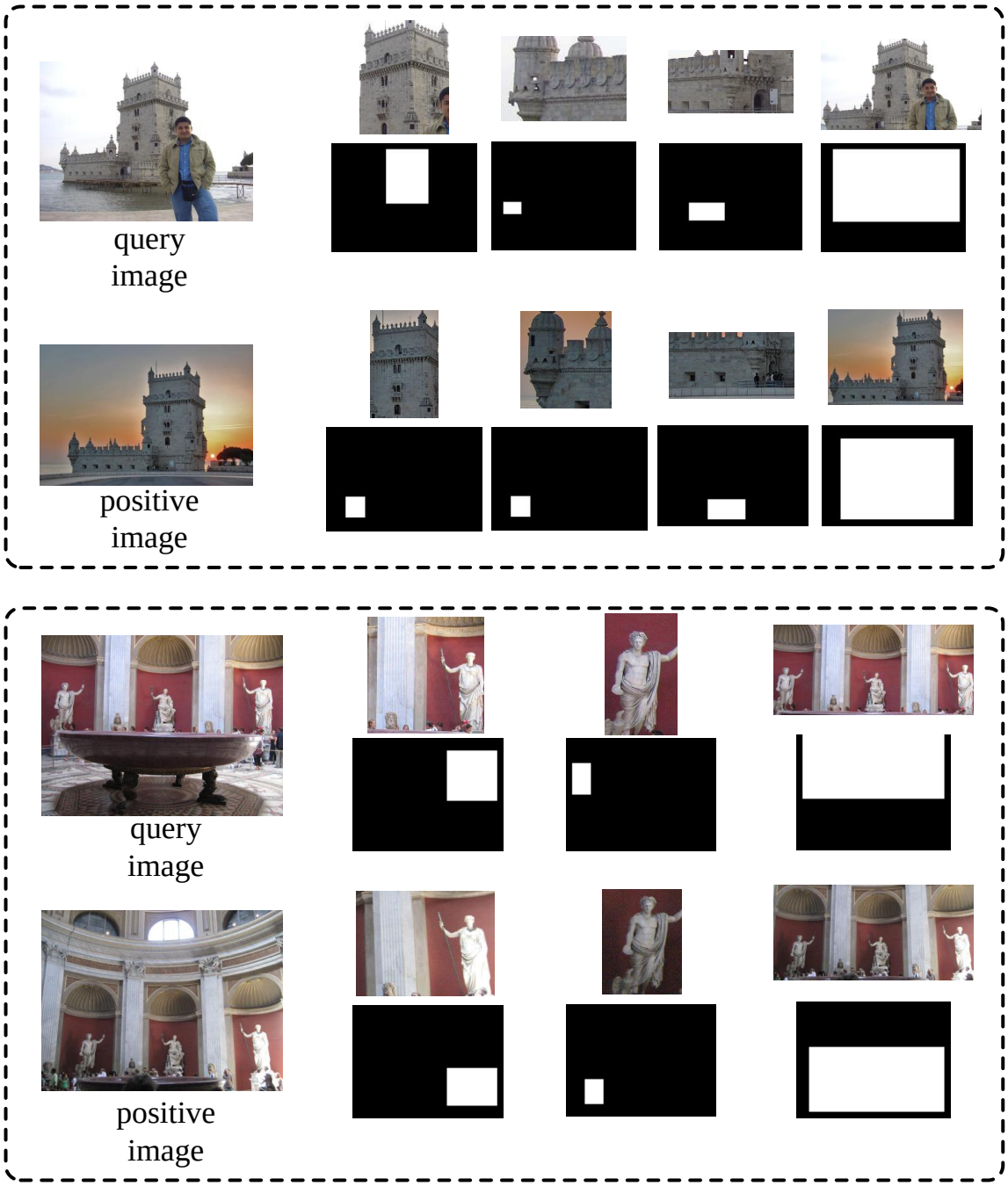


Figure 3.5: Examples of generated training data.

and corresponding ground-truth attention maps $\mathbf{I}_q$ and $\hat{\mathbf{A}}$, are generated, as described in Section 3.3.1. These query regions and ground-truth attention maps are then used for training. When considering each negative image pair, it is enforced that $\mathbf{I}_q$, defined through positive matches, would not match any region within the negative image. In this case, $\hat{\mathbf{A}}_{neg} \equiv \mathbf{0}$.

Let $\mathbf{A}$ denote the generated co-attention map of the candidate image $\mathbf{I}_c$ conditioned on the content from the query $\mathbf{I}_q$. $\hat{\mathbf{A}}$ denotes the ground-truth co-attention map and has been interpolated to be of the same size as $\mathbf{A}$. The loss function for training the CANet is represented by the mean square error (MSE) between the two maps:

$$\mathrm{MSE}(\mathbf{A}, \hat{\mathbf{A}}) = \frac{1}{L} \sum_{l \in L} \|\mathrm{A}_l - \hat{\mathrm{A}}_l\|^2, \tag{3.1}$$

where L represents all locations on the generated co-attention map and $\mathrm{A_l}$ is the attention map value at location $l \in L$. This loss function would enable the CANet to learn the matching regions from image pairs and then to use such matching knowledge for co-attention generation and feature re-weighting at the retrieval stage, as described in the next section. We consider mean square error loss instead of the cross entropy loss because, according to our tests, applying MSE loss would make the final generated co-attention map more accurate, with fewer uncertainties and better contrast around the target objects.

3.4 Embedding co-attention into the CBIR pipeline

The proposed Conditional Attention Network (CANet), described in Section 3.3, represents an independent co-attention generator module that can be embedded into the existing deep learning CBIR feature extraction pipeline. As spatial pooling has been successfully used for feature extraction from images in general applications and the Generalized Mean pooling (GeM) [98] provides state-of-the-art performance for image retrieval, the GeM model from [98] is considered as the baseline model. In the following, the details of the baseline GeM model are introduced, and how to embed the co-attention feature map generated by CANet into the GeM feature extraction pipeline is explained.

Baseline GeM model. The original GeM feature extraction pipeline, proposed in [98],

consists of 3 parts: a fully convolutional backbone network for feature tensor extraction, a generalized mean pooling layer to transform the feature tensor into a compact feature vector and a whitening layer for feature normalization. The off-shelf pre-trained GeM model from [98] is used as the baseline CBIR model to more clearly show the relevance of the proposed CANet on the retrieval performance. It needs to point out that there are two different versions of the GeM pooling implementation provided by the authors. In the original GeM paper [98], the author states that the whitening module works better when it is learned and applied as a post-processing module after the backbone network has been trained. However, according to the results and description from the author's latest code release page[6], implementing the feature whitening by a fully connected layer and training it jointly with the backbone network could lead to better retrieval results. In this work, both versions of the GeM with different backbone structures are tested. Retrieval results when using GeM with and without CANet are provided in Section 3.5.

Combining the co-attention map with GeM pooling. At the retrieval stage, assuming that the global feature vector $\mathbf{V}_q$ of the query image $\mathbf{I}_q$ is extracted by the original GeM model. The feature tensor $\mathbf{X}_c \in \mathbb{R}^{H_c \times W_c \times D}$ of the candidate image $\mathbf{I}_c$ is extracted by the GeM backbone network, where H_c, W_c, D represent the height, width and dimension (channel count) of the feature tensor, respectively. The co-attention map of $\mathbf{I}_c$, under the condition of the query $\mathbf{I}_q$ generated by CANet is denoted by $\mathbf{A}$ and has been interpolated to the size of $H_c \times W_c \times 1$. The feature tensor $\mathbf{X}_c$ is location-wise re-weighted by the co-attention map $\mathbf{A}$ followed by GeM pooling to get the final co-attention weighted GeM feature vector $\mathbf{V}_c$ for the candidate image $\mathbf{I}_c$, as illustrated in Figure 3.6. The final similarity measure between $\mathbf{I}_q$ and $\mathbf{I}_c$ is performed by using the cosine similarity between $\mathbf{V}_q$ and $\mathbf{V}_c$.

In practice, the actual output attention score is not likely to cover the whole range of the Sigmoid function output of $(0, 1)$. Instead, it would be concentrated within a specific localized range $[a, b]$, with $a > 0$ and $b < 1$, leading to an insignificant difference between attention scores of the foreground region of interest and that of the background noises. Accordingly, before re-weighting the candidate image feature tensor $\mathbf{X}_c$, values of the co-attention map output by the Sigmoid function are stretched and re-normalized by the

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

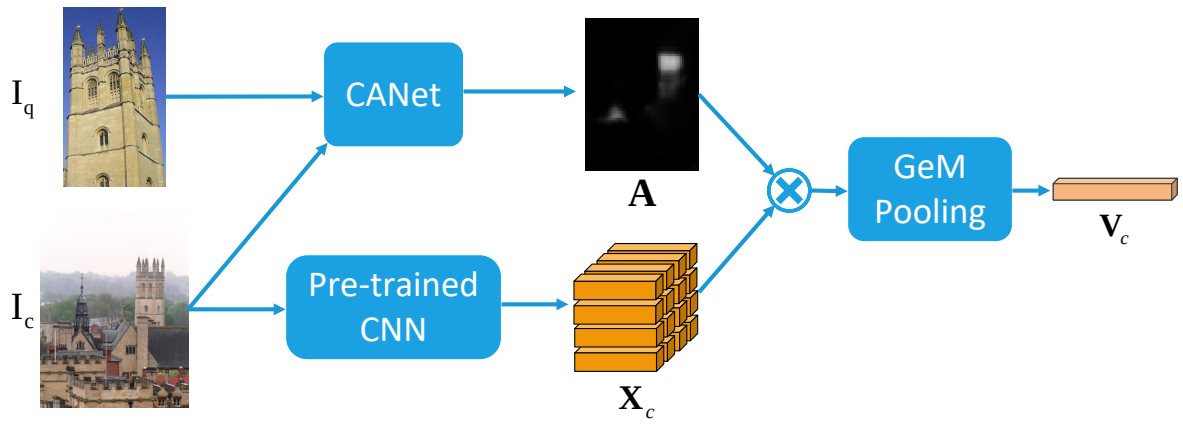


Figure 3.6: Embedding the co-attention map into GeM feature extraction.

min-max normalization:

$$\mathbf{A}' = \frac{\mathbf{A} - A_{\min}}{A_{\max} - A_{\min}}, \tag{3.2}$$

where $\mathbf{A}$ represents the original co-attention map output by the Sigmoid function, $\mathrm{A}_{min} = \min(\mathbf{A})$ and $\mathrm{A}_{max} = \max(\mathbf{A})$ are the minimum and maximum values in $\mathbf{A}$.

Multi-scale scheme for co-attention generation. The multi-scale feature extraction scheme has been widely applied in image feature extraction for CBIR. In the original GeM pipeline, global feature vectors from different input images, of various scales, are fused by average pooling, then L2 normalized.

At the retrieval stage, in order to obtain more accurate co-attention maps, a multi-scale scheme is also applied for the co-attention map generation before combining it with GeM features [98]. As shown in Figure 3.7, the query image $\mathbf{I}_q$ is fed into the CANet, together with the candidate image $\mathbf{I}_c$ represented at several different scales, while preserving the initial aspect ratio. All attention maps generated at different scales are resized to the same resolution and then weighted summed before the min-max normalization. The weights are evaluated by applying max pooling and the Softmax activation function on the co-attention maps from different candidate image scales.

Re-ranking with query expansion. Query expansion has been widely used for improving image retrieval results [7, 124, 60]. In the CANet enabled CBIR pipeline, the α-weighted query expansion (αQE) [98] is applied for retrieval result re-ranking. αQE acts on the feature vectors of top-ranked nQE images from the initial retrieval result by applying weighting averaging and re-normalization. The weight of the i-th ranked image

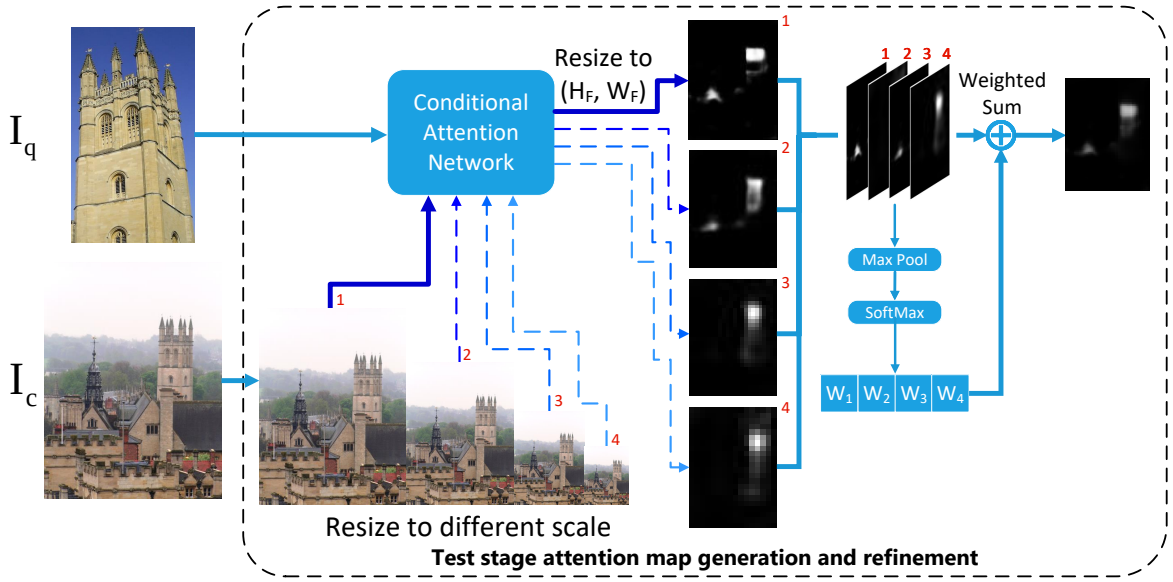


Figure 3.7: Attention map generation and refinement with the multi-scale scheme during the retrieval (testing) stage.

descriptor is defined by $(\mathbf{V}_q^\top \mathbf{V}_i)^\alpha$ where $\mathbf{V}_q$ and $\mathbf{V}_i$ are the feature vectors corresponding to the query image and the i-th ranked retrieval image. The aggregated feature vector serves as a query descriptor for a second-round retrieval test and produces the final retrieval result.

3.5 Experiments

3.5.1 Evaluation datasets

For retrieval performance evaluation, two benchmark datasets: Oxford5k [93] and Paris6k [94] (Oxf/Par) are considered. Oxford5k contains 5062 images which are collected from Flickr with 17 tags of buildings from Oxford. All images are manually annotated. Paris6k, consisting of 6412 images, is also collected from Flickr by searching for 12 Paris landmark tags. Both of them provide 55 images as queries for the image retrieval evaluation. Oxford105k and Paris106k are expanded versions of the Oxford5k and Paris6k by adding 100K distractor images from Flickr. In addition, ROxford5k and RParis6k (ROxf/RPar) are revisited versions of Oxford5k and Paris6k. Each contains 15 extra new challenging queries and re-arrange the potential positive images of each query into three groups:*Easy, Medium, Hard*, corresponding to different difficulty levels.

All these evaluation datasets provide bounding boxes for each query image, outlining the object of interest. Following the standard evaluation protocol, each query image is cropped with its bounding box. The cropped query image serves as input for both the GeM model and the proposed CANet. During the evaluation, all input images are limited to a maximum size of 1024×1024 pixels. The mean average precision (mAP) [93] is used as a performance measure for the results on all datasets.

3.5.2 Implementation details

The backbone network of the proposed CANet, considered as VGG16 [111], is pre-trained on ImageNet [103] for classification as up-stream task training. During the training stage, all input images are resized to a maximum of 362×362 pixels while keeping the original image ratio. After processing by VGG16 and being down-sampled four times by four max-pooling layers, where each max-pooling layer would reduce the size of its input by half, the output co-attention map has a maximum size of 22×22 ($\frac{362}{16} \approx 22$). The ground-truth attention map's size is supposed to equal that of the generated attention map for the mean square error calculation. By taking all these aspects into consideration, the long side of both key-point maps $\mathbf{M}_Q$ and $\mathbf{M}_P$ is set to be 22 while keeping the original image's aspect ratio for training data generation. The CANet is trained with the Adam optimizer [62], using an initial learning rate $l_0 = 10^{-4}$, momentum of 0.9 and weight decay of 5×10^{-4}. A cosine learning rate decay strategy is applied, and the training is performed for 100 epochs. For each epoch, 2000 image tuples are randomly selected from the rSfM-120k [98] dataset with a batch size of 5 tuples.

At the evaluation stage, 3 scales: $\{1, \sqrt{2}, \frac{1}{\sqrt{2}}\}$ are considered for both GeM feature extraction and co-attention generation. For the re-ranking with query expansion, nQE = 10 for Oxford, nQE = 50 for Paris dataset and $\alpha = 3$, as explained in Section 3.4.

3.5.3 Co-attention generation results

Figure 3.8 shows some examples of generated attention maps when considering various pairs of candidate and query images. Examples 1-4 show that the proposed co-attention model can accurately locate the target object under various challenging situations, such

as when the images are characterized by different acquisition parameters, changes in the light condition or when the object of interest is small and far away. Examples 5 and 6 show the generated attention map for the same candidate image but consider different query images. Unlike the wGeM failure example, shown in Figure 1.3, CANet can correctly highlight the target query object based on the input query image, even when there are two potential objects of interest in the same image. Example 7 shows how the proposed conditional attention model works with unseen image content during the training. Although the network was trained with landmark building images from rSfM-120k dataset [97, 98], which displays architecture buildings, it is also sensitive to human face content and highlights corresponding regions. However, it can not distinguish different human faces but just uniformly highlight all potential match regions. Example 8 shows a case where the CANet fails. Because the global pooling is used to extract the feature from the query image, the retrieval could fail if it contains too much distraction content. As shown in example 9, if the target pyramid is manually cropped out and all background is masked by 0, the quality of the generated attention map is improved.

3.5.4 Image retrieval results

Impact of CANet. The retrieval results when combining off-shelf pre-trained GeM models with the co-attention generated by CANet are presented in Table 3.1. In total, four off-the-shelf pre-trained GeM models are provided in [98]. They have either different ways of feature whitening or different backbone structures. The re-ranking with query expansion using α-weighted query expansion (αQE) [98] is also considered. According to the results from Table 3.1, no matter which one of the pre-trained GeM models, embedding co-attention generated by the proposed CANet into the GeM pipeline can stably improve the original model's performance.

Comparison with other works. The retrieval results comparison of the proposed method (GeM+CANet) and other existing works on Oxford5k, Paris6k, Oxford105k and Paris106k datasets are shown in Table 3.2. When VGG16 [111] is used as the backbone network of GeM, GeM+CANet outperforms other methods shown in the table. When ResNet101 [45] is implemented as the backbone network of GeM and combined with αQE re-ranking with query expansion [98], GeM+CANet+αQE provides new state-of-the-art

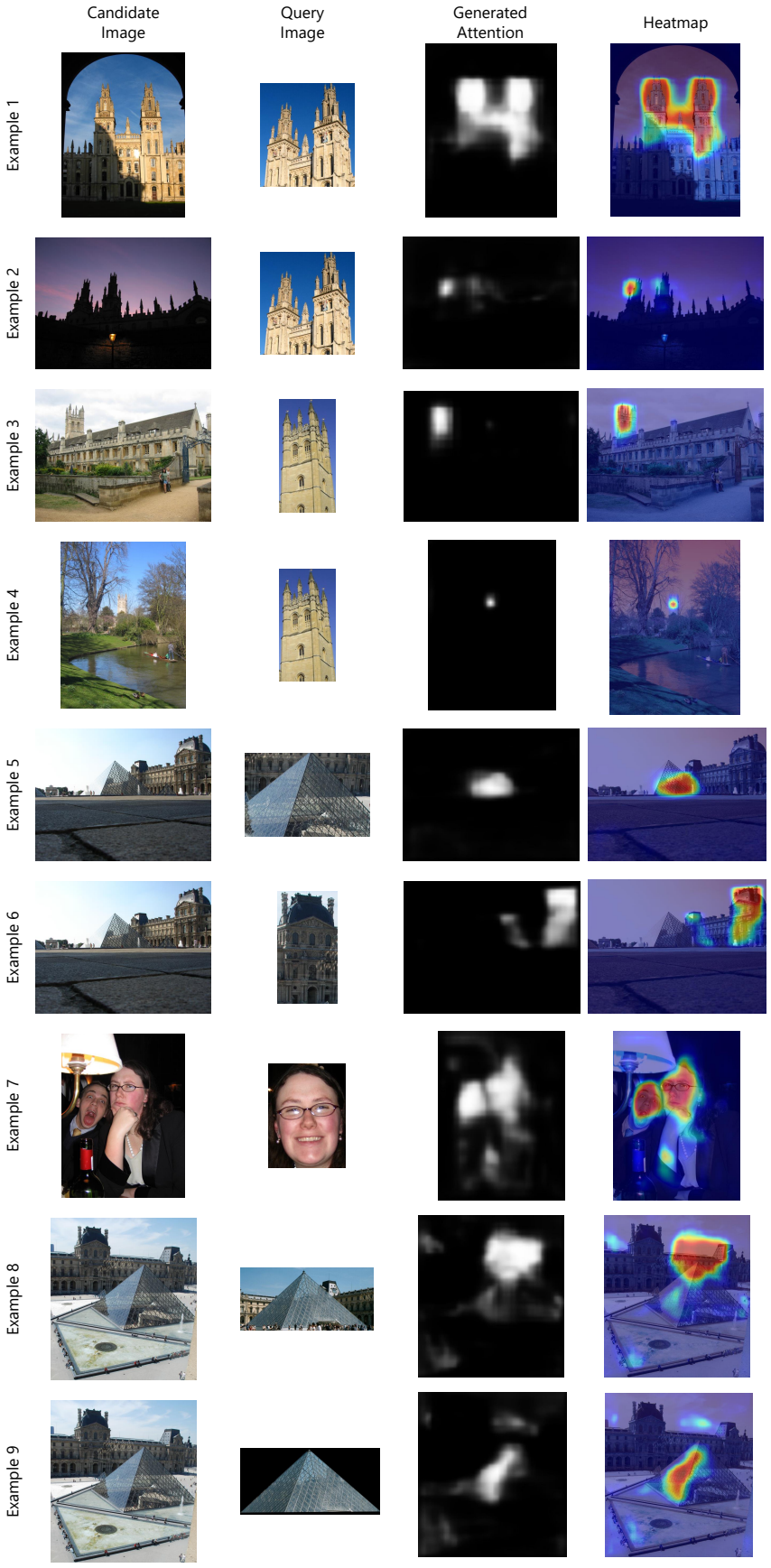


Figure 3.8: Attention map results for the proposed conditional attention model. Candidate images and the query images are displayed in the first and second rows, respectively. Third and fourth rows represent the generated attention maps and corresponding heatmaps, after min-max normalization and up-sampling to the original image size.

Method	Backbone	Whiten	Oxford5k	Oxford105k	Paris6k	Paris106k
(A) Off-shelf GeM without re-ranking						
GeM [98]	VGG16	Lw	87.9	83.3	87.7	81.3
	Res101	Lw	87.8	84.6	92.7	86.9
	Res50	Fw	86.9	83.8	90.8	85.8
	Res101	Fw	88.4	85.5	92.3	87.0
(B) Off-shelf GeM + CANet without re-ranking						
GeM+CANet	VGG16	Lw	89.1	85.8	90.7	85.9
	Res101	Lw	89.6	86.7	93.2	88.5
	Res50	Fw	89.6	87.4	93.3	88.6
	Res101	Fw	91.0	88.6	94.7	90.6
(C) Off-shelf GeM with QE re-ranking						
GeM+αQE [98]	VGG16	Lw	91.9	89.6	91.9	87.6
	Res101	Lw	91.0	89.5	95.5	91.9
	Res50	Fw	90.8	89.9	92.8	89.2
	Res101	Fw	92.0	91.0	94.2	92.0
(C) Off-shelf GeM + CANet with QE re-ranking						
GeM+CANet+αQE	VGG16	Lw	93.0	90.5	93.0	88.6
	Res101	Lw	92.4	89.6	96.1	92.1
	Res50	Fw	93.4	91.8	94.3	91.0
	Res101	Fw	93.6	92.0	96.5	93.5

Table 3.1: Image retrieval performance using the mean average precision (mAP) when considering different off-shelf pre-trained GeM models. The column "Whiten" indicates the way of feature whitening implementation. "Lw" means the feature whitening is applied as a learned post-processing module while "Fw" means the feature whitening is applied as an end-to-end fully trained connected layer.

results on these four datasets.

The retrieval results comparison of our method GeM+CANet and other works on ROxf/RPar datasets are shown in Table 3.3. It can be observed that even on the *hard* set of ROxf/RPar dataset, the proposed method GeM+CANet still gives the best retrieval results. The only exception is on the *hard* set of ROxf/RPar when compared to DELF-HQE+SP with query expansion. However, DELF-HQE+SP utilizes the local features for second-round re-ranking. It is trained on the Google Landmark dataset [87], which is a much larger dataset than the rSfM-120k dataset used for training the CANet and the baseline GeM model. Even under these circumstances, the proposed method still shows comparable results and outperforms the other methods when considering mAP in the top 10 results.

Method	Backbone	Oxford5k	Oxford105k	Paris6k	Paris106k
(A) No re-ranking					
BoW-CNN [81]	VGG16	73.9	59.3	82.0	64.8
NetVLAD [4]	VGG16	71.6	–	79.7	–
SPoC [7]	VGG16	68.1	61.1	78.2	68.4
CroW [60]	VGG16	70.8	65.3	79.7	72.2
R-MAC [40]	VGG16	83.1	78.6	87.1	79.7
GeM [98]	VGG16	87.9	83.3	87.7	81.3
GeM+CANet*	VGG16	89.1	85.8	90.7	85.9
R-MAC [40]	Res101	86.1	82.8	94.5	**90.6**
WGeM [138]	Res101	88.8	85.6	92.5	–
GeM [98]	Res101	87.8	84.6	92.7	86.9
GeM+CANet*	Res101	**91.0**	**88.6**	**94.7**	**90.6**
(B) re-ranking with query expansion (QE)					
CroW+QE [60]	VGG16	74.9	70.6	84.8	79.4
BoW-CNN+QE [81]	VGG16	78.8	65.1	84.8	64.1
R-MAC+QE [40]	VGG16	89.1	87.3	91.2	86.8
GeM+αQE [98]	VGG16	91.9	89.6	91.9	87.6
GeM+CANet+αQE*	VGG16	93.0	90.5	93.0	88.6
R-MAC+QE [40]	Res101	90.6	89.4	96.0	93.2
WGeM+QE [138]	Res101	91.7	89.7	96.0	–
GeM+αQE [98]	Res101	91.0	89.5	95.5	91.9
GeM+CANet+αQE*	Res101	**93.6**	**92.0**	**96.5**	**93.5**

Table 3.2: Image retrieval performance (mAP) comparison on Oxford5k, Oxford105k, Paris6k and Paris106k dataset. "*" marks the proposed method. The highest mAP score is highlighted in bold.

3.6 Ablation study and discussion

This section presents the results of ablation experiments to show the impact of different hyper-parameter settings on retrieval performance. By default, the off-shelf GeM model, with VGG16 as the backbone and whitening applied by a learned post-processing module, serves as the baseline model.

Method	Backbone	ROxford5k				RParis6k			
		Medium		Hard		Medium		Hard	
		mAP	mAP@10	mAP	mAP@10	mAP	mAP@10	mAP	mAP@10
(A) No re-ranking									
SPoC	VGG16	38.0	54.6	11.4	20.9	59.8	93.0	32.4	69.7
CroW	VGG16	41.4	58.8	13.9	25.7	62.9	94.4	36.9	77.9
NetVLAD	VGG16	37.1	56.5	13.8	23.3	59.8	94.0	35.0	73.7
MAC	VGG16	58.4	81.1	30.5	48.0	66.8	97.7	42.0	82.9
GeM	VGG16	61.9	82.7	33.7	51.0	69.3	97.9	44.3	83.7
GeM+CANet*	VGG16	66.0	86.9	39.0	56.4	73.2	99.1	49.2	87.0
DELF–ASMK+SP	Res50	67.8	87.9	43.1	62.4	76.9	99.3	55.4	93.4
GeM+CANet*	Res50	70.8	89.6	45.5	62.7	78.9	99.3	58.5	92.0
SPoC	Res101	39.8	61.0	12.4	23.8	69.2	96.7	44.7	78.0
CroW	Res101	42.4	61.9	13.3	27.7	70.4	97.1	47.2	83.6
R-MAC	Res101	60.9	78.1	32.4	50.0	78.9	96.9	59.4	86.1
GeM	Res101	64.7	84.7	38.5	53.0	77.2	98.1	56.3	89.1
GeM+CANet*	Res101	**72.2**	**91.0**	**46.8**	**64.3**	**80.3**	**99.1**	**60.9**	**93.3**
(B) re-ranking with query expansion (QE)									
GeM+αQE	VGG16	66.6	85.7	38.9	57.3	74.0	98.4	51.0	88.4
GeM+CANet+αQE*	VGG16	73.2	88.3	44.4	63.3	78.3	99.4	56.3	91.4
DELF-HQE+SP	Res50	73.4	88.2	**50.3**	67.2	84.0	98.3	**69.3**	93.7
GeM+CANet+αQE*	Res50	75.6	92.0	48.6	68.1	84.3	**99.6**	68.7	94.0
R-MAC+αQE	Res101	64.8	78.5	36.8	53.3	82.7	97.3	65.7	90.1
GeM+CANet+αQE*	Res101	**76.3**	**92.3**	49.9	**70.0**	**84.5**	99.3	66.5	**94.7**

Table 3.3: Image retrieval performance (mAP) comparison on Oxford5k, Oxford105k, Paris6k and Paris106k dataset. "*" marks the proposed method. All mAP results of existing works are provided by [96]. The highest mAP score is highlighted in bold and mAP@10 indicates that the mAP is calculated on top 10 results.

3.6.1 Impact of re-normalization

As described in Section 3.4, min-max normalization is applied, according to Eq. (3.2), to enhance the contrast in the attention results over the entire image. Figure 3.9 shows the co-attention result with and without the min-max normalization for a pair of query and candidate images showing a building from Oxford. Before the min-max normalization, the target tower from the candidate image is not well emphasized when compared with the heat-map resulting after applying min-max normalization in the fourth image from Figure 3.9. The min-max normalization enhances the contrast between the foreground target object and the irrelevant regions, even when multiple similar architectural features are present in the candidate image. Table 3.4 contains the statistical retrieval results with and without the min-max normalization. Although without the min-max normalization, the CANet can still improve the original GeM model's performance. By applying it, the image retrieval performance (mAP) on Oxford5k (Paris6k) is further improved by 0.3% (1.6%).

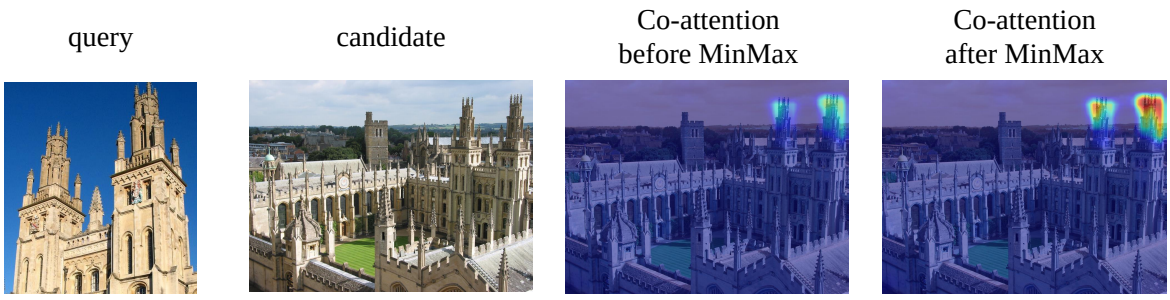


Figure 3.9: Co-attention map visualization with and without min-max normalization.

Method	GeM Backbone	min-max	Oxford5k	Paris6k
GeM+CANet*	VGG16	✗	88.8	89.1
GeM+CANet*	VGG16	✔	89.1	90.7
baseline GeM [98]	VGG16	-	87.9	87.7

Table 3.4: Image retrieval performance (mAP) comparison when considering the min-max re-normalization from Eq. 3.2 and without.

3.6.2 Impact of the multi-scale scheme

Figure 3.10 shows some co-attention examples when using a single, three or five scaling factors for the same query as in Figure 3.9, but considering a candidate image under utterly different lighting conditions. It can be observed that implementing the co-attention generation with three scales can greatly improve the accuracy and quality of generated co-attention map when compared to using a single scale. However, using five scales does not make much difference. Table 3.5 statistically compares the image retrieval performance when combining the baseline GeM model with co-attention maps from different input image scale settings. Combining GeM with the co-attention map from a single scale could only bring minimal improvement. However, using five scales would not bring significant additional improvement while requiring more computation costs. The default setting of the three scales reaches a good balance between the computation cost and performance improvement.

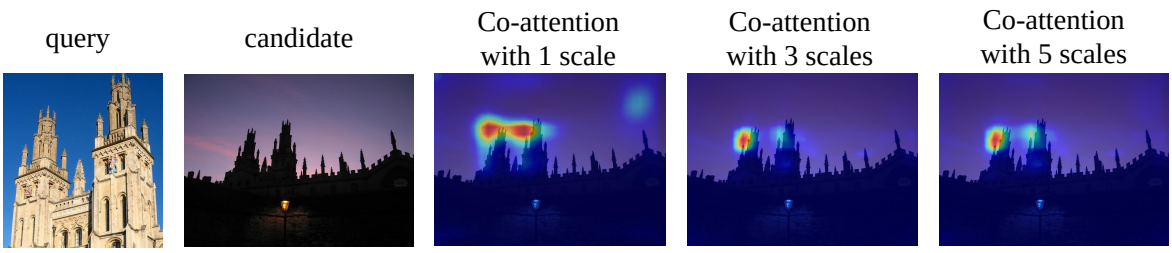


Figure 3.10: Co-attention map visualization with different input candidate image scales.

Method	GeM Backbone	Co-attention Scales					Oxford5k	Paris6k
		1	$\frac{1}{\sqrt{2}}$	$\sqrt{2}$	$\frac{1}{2\sqrt{2}}$	$\frac{1}{2}$		
GeM+CANet*	VGG16	✔	-	-	-	-	88.6	90.0
GeM+CANet*	VGG16	✔	✔	✔	-	-	89.1	90.7
GeM+CANet*	VGG16	✔	✔	✔	✔	✔	89.2	90.5
baseline GeM [98]	VGG16	-	-	-	-	-	87.9	87.7

Table 3.5: Image retrieval performance (mAP) when considering different image scales of co-attention generation.

3.6.3 Impact of CANet backbone structure

The impact caused by the choice of CANet backbone structure is explored. Apart from the default VGG16 backbone structure, ResNet50 and ResNet101 are also tested as backbone networks. Figure 3.11 presents the co-attention visualization with different backbone structures for CANet. All these networks correctly highlight the target object. An interesting observation is that the co-attention generated with the shallowest VGG16 structure tends to uniformly highlight the whole target building. As the depth of the backbone network increases from VGG16 to ResNet101, the co-attention tends to more and more focus on the specific representative part of the building. For example, the co-attention with ResNet101 would lay much more emphasis on the sharp top part of the target building than the co-attention with VGG16. Table 3.6 provides the retrieval performance of GeM+CANet when considering different backbone network structures for the CANet. Deeper networks usually provide a more comprehensive feature extraction, leading to better results. However, according to the results in Table 3.6, using ResNet101 or ResNet50 as a backbone for CANet would not further improve the retrieval performance. Considering the co-attention visualization from Figure 3.11, one possible reason could be that deep backbone networks, like ResNet101/ResNet50 make the co-attention too focused on the local part of the target object, ignoring the contextual content.

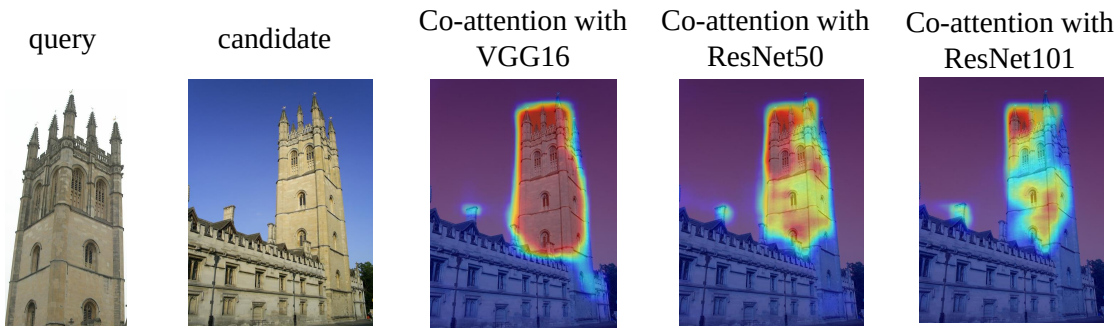


Figure 3.11: Co-attention map visualization when considering different backbone network structures.

Method	GeM Backbone	CANet Backbone	Oxford5k	Paris6k
GeM+CANet*	VGG16	VGG16	89.1	90.7
GeM+CANet*	VGG16	ResNet50	88.5	89.0
GeM+CANet*	VGG16	ResNet101	88.6	89.7
baseline GeM [98]	VGG16	-	87.9	87.7

Table 3.6: Image retrieval performance (mAP) when considering different backbone structures for CANet.

3.6.4 Impact of feature fusion module

Count of multi-scale block. As illustrated in Figures 3.1(c) and 3.7, the proposed CANet utilizes stacks of multi-scale blocks for feature fusion. Results when considering various numbers of scaling blocks are provided in Figure 3.12 when considering a church tower from Oxford as a query image. Considering stacks of multi-scale blocks can lead to wider receptive fields and better feature detection under perspective projection changes leading to comprehensive feature object or region representation. From Figure 3.12, it can be observed that employing 3 multi-scale blocks leads to the most comprehensive co-attention map, in which the whole target object is encompassed. Otherwise, the co-attention maps tend to be disconnected. Quantitative retrieval results when varying the number of multi-scale block[7] are provided in Table 3.7. As we can see, it gives the best mAP results on both Oxford5k and Paris6k datasets when considering 3 multi-scale blocks.

Convolution with dilation. The impact that could be caused by the dilation of con-

[7]When reducing the number of multi-scale blocks, they are replaced by simple convolution layers with kernel size of 3×3 while keeping the input and output channel count.

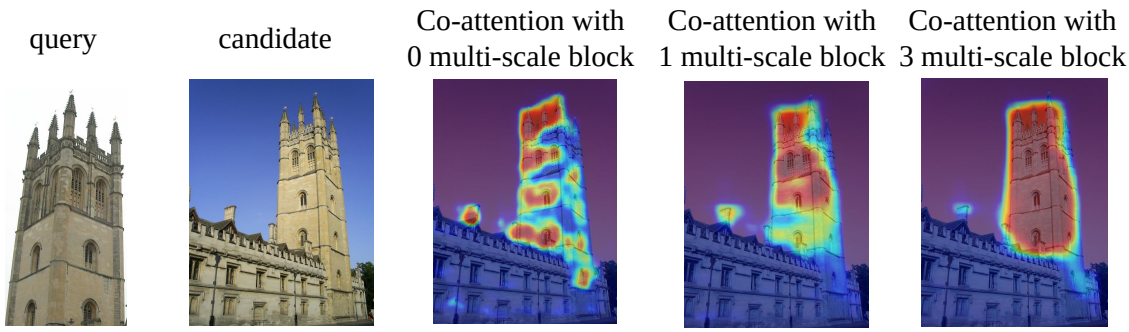


Figure 3.12: Co-attention map visualization when considering between zero and three multi-scale blocks.

Method	GeM Backbone	No. of Multi-scale Blocks	Oxford5k	Paris6k
GeM+CANet*	VGG16	0	88.5	90.0
GeM+CANet*	VGG16	1	88.9	90.4
GeM+CANet*	VGG16	3 (default)	89.1	90.7
baseline GeM [98]	VGG16	-	87.9	87.7

Table 3.7: Image retrieval performance (mAP) when varying the number of Multi-scale blocks in the CANet backbone structure.

volution layers is also tested. Dilation [148] is a technique that extends the convolution kernel size by inserting zero weights among the elements of the extended kernel. This leads to a wider receptive field of each convolution layer without significantly increasing the computation cost. When considering convolution with dilation, the convolution layers of kernel sizes 5×5 and 7×7 in all multi-scale blocks are replaced by convolution layers of kernel size 3×3 but with dilation rates of 2 and 3 respectively. The results provided in Figure 3.13 qualitatively compare the co-attention maps with and without convolution dilation. When can observe that when considering kernel dilation, CANet can still correctly highlight the target building, but it does not cover the whole salient tower from the query as the co-attention generated by the kernel without dilation. Table 3.8 provides the mAP retrieval results of GeM+CANet with and without dilation. Applying the fusion module with dilation could still improve the baseline GeM model's performance. However, the improvement is not as great as considering convolution kernels without dilation.

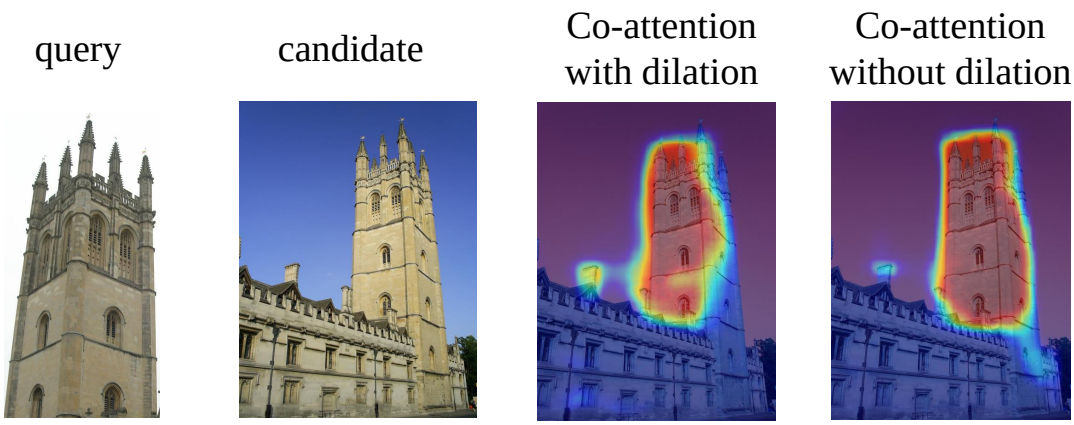


Figure 3.13: Co-attention map visualization with and without convolution dilation.

Method	GeM Backbone	Convolution dilation	Oxford5k	Paris6k
GeM+CANet*	VGG16	✗	89.1	90.7
GeM+CANet*	VGG16	✓	88.8	90.1
baseline GeM [98]	VGG16	-	87.9	87.7

Table 3.8: Image retrieval performance (mAP) when considering convolution dilation for the feature fusion module.

3.6.5 Using key-point matching as co-attention

CANet is trained with data resulting from the key-points detected by the SuperPoint model. In other words, CANet training can be seen as being supervised by the features, representing the output of the SuperPoint model. In the following, it is considered to directly apply matching key-points resulting from the SuperPoint output and use these as co-attention to re-weight the GeM features. Generally speaking, there are two main drawbacks of directly using SuperPoint based key-point matching as co-attention for feature extraction and re-weighting. Firstly, SuperPoint is initially trained with code vectors defined by constraints of simple shapes, such as triangles, quadrilaterals, polygons and so on. Accordingly, SuperPoint detection is sensitive to corners, vertices or high-contrast edges of objects. In real-world images, SuperPoint works well with dense, complex textured areas but would not identify matching points on large smooth areas with no or minor variation. In example 1 from Figure 3.14, there are many dense, complex textures from historic buildings, resulting in multiple correct matching key-points detected by the SuperPoint model. However, in example 2, the query object has a relatively simpler body texture. We can observe that only a few key points are detected, which are mainly distributed over

the upper structure of the tower. In a way, simple SuperPoint based key-point matching has poor generalization ability when used for real-world CBIR tasks. Secondly, the SuperPoint model only performs point-level matching. In other words, it does not consider the higher-level semantic meaning and is likely to mismatch in certain circumstances. From example 2 of Figure 3.14, it can be seen that there are some mismatching key-point pairs in the third column of images under "match line". The SuperPoint model considers that the edge points from the windows of the query tower building match with the window from the bottom area of the tower from the candidate image. If visualizing these matching points based on the local match scores output by SuperPoint model, as shown in the images from the fourth column of Figure 3.14, the resulting co-attention is not only very sparse and too localized but also incorrect. In Table 3.9 we provide statistical retrieval results when considering the co-attention based on using the proposed CANet and when directly applying the SuperPoint. We can observe that CANet significantly improves the original GeM model's performance when directly using SuperPoint to guide the feature re-weighting for CBIR.

In summary, applying the data generation described in Section 3.3 and training CANet for the co-attention generation achieves better generalization ability, resulting in more comprehensive co-attention maps for CBIR feature re-weighting.

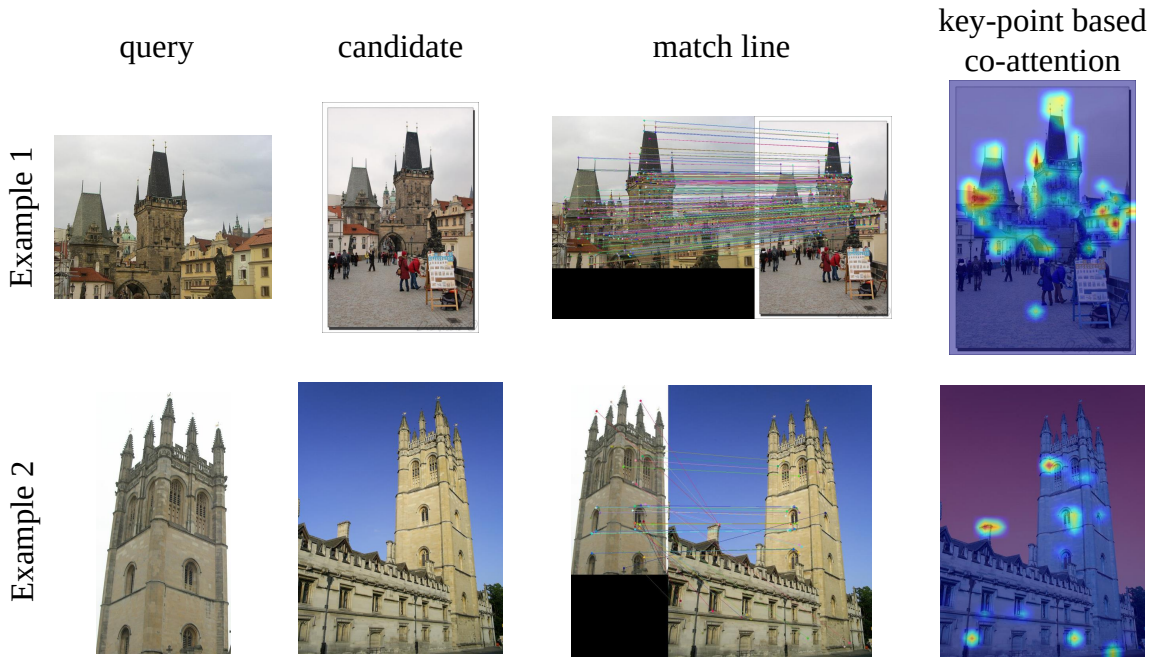


Figure 3.14: Key-point match and key-point based co-attention visualization.

Method	GeM Backbone	Co-attention	Oxford5k	Paris6k
GeM+CANet*	VGG16	SuperPoint	87.2	85.3
GeM+CANet*	VGG16	CANet	89.1	90.7
baseline GeM [98]	VGG16	-	87.9	87.7

Table 3.9: Image retrieval performance (mAP) when considering the SuperPoint directly as co-attention.

3.6.6 Computation cost

Considering a pair of the query image and candidate image of size 512×512 in original scale as input, ResNet101(R101) [45] as the backbone network for GeM feature extraction, VGG16 [111] as the backbone network for the CANet, Table 3.10 (a) and (b) compare the time cost of the GeM model at the online retrieval stage with or without the proposed CANet. All experiments were performed 100 times, and we report the average time cost in Table 3.10.

As illustrated in Figure 1.2, candidate images' features are supposed to be pre-cached at the offline stage, what we need to do at the online stage is extract the query image's feature and perform similarity measure. As shown in Table 3.10 (b), following the standard CBIR pipeline, the original GeM model only takes around 16ms to get the final match score between one pair of the query image and candidate image.

However, when considering the proposed CANet, as the convolution feature tensor of the candidate image needs to be re-weighted by the co-attention map before pooling, the feature extraction of the candidate image and the co-attention map generation both need to be conducted at the online stage. As shown in Table 3.10 (a), the extra time cost of CANet is mainly caused by candidate feature extraction and forward through the CANet structure. Compared with the original GeM pipeline, it takes around extra 21ms for one candidate image.

3.6.7 Limitations and future work

Although the proposed CANet can be used to generate good co-attention maps and improve the original GeM model performance, as discussed in Section 3.6.6, a major draw-

For query feature extract		For candidate feature extract	For CANet			Cosine
Backbone(R101)	Pool and whiten	Backbone(R101)	Backbone(VGG16)	Fusion module	Re-weight and pool	similarity
16	0.2	16	3.4	2.4	0.27	0.05

(a) Time cost of each component within the proposed clustering-based co-attention pipeline at the online evaluation stage.

For query feature extract		Cosine
Backbone(R101)	Pool and whiten	similarity
16	0.2	0.05

(b) Time cost of each component within the original GeM [98] pipeline at the online evaluation stage.

Table 3.10: Time cost analysis of each component within the CANet pipeline (a) and the GeM pipeline (b) at the online evaluation stage. A pair of one query image and one candidate image serves as input. The time cost of each component is reported in milliseconds (ms).

back of the proposed CANet is the computation cost caused by the query sensitivity. As each candidate image's global feature vector needs to be built under the condition of the input query image, feeding the candidate image through the CANet and GeM model both have to be performed at the online retrieval stage. Given one query image and process candidate images in batch-wise manner, it takes around one hour to search on the Oxford5k or Paris6k dataset. The retrieval time cost is linearly increased to about one day when considering the 100k distractor set. Addressing extra computation costs caused by query sensitivity would be the next step in this research direction.

3.7 Conclusion

In this research study, an independent conditional attention model is proposed for content-based image retrieval, which does not require any manual annotations for training. Instead, the model is trained on automatically generated training data by finding correspondences from existing matching image pairs, using pre-trained SuperPoint model. As shown in the experiments, the proposed attention model can accurately highlight the region, matching the content of the query image onto the candidate image. It performs well even under various challenging situations such as when significantly changing the illumination conditions or when the image acquisition parameters change significantly under different perspective projection conditions. When combined with the GeM feature extraction method, the proposed methodology achieves state-of-the-art image retrieval results

on Oxford5k and Oxford105k datasets. However, the huge extra computation cost caused by the query-sensitivity could be a critical problem for the proposed CANet, especially when considering large-scale image retrieval. How to further simplify the co-attention generation procedure, reducing the cost at online retrieval stage is the central question of next work.

Regarding the experimental results of the CANet, we can draw some conclusions. First, as CANet utilises stacks of convolution layers to fuse the query global feature and each candidate image local feature, each co-attention score on the resulting co-attention map could be treated as deriving from interaction between the query global feature and corresponding candidate image local feature. In other words, this global-to-local framework, in which the query global feature is considered to each candidate image's local feature, could work as an effective manner for attention generation. Even though they may contain unequal amounts of information, as the query global feature contains the information of the whole query image while each entry on the candidate image feature tensor only contains features from the neighbour region. Second, as discussed in the Section 3.6.5, although the CANet is under the supervision of the SuperPoint model, the CANet still leads to better co-attention generation as well as better final retrieval results than directly utilising the key-point output by SuperPoint as co-attention. Even though the training data generation pipeline sometimes could only generate patch-level matches as training data, as shown in Figure 3.5. The training procedure seems to enable CANet with better generalisation ability and accuracy than the original SuperPoint model. Third, according to the ablation study in Section 3.6.4, the number of multi-scale blocks greatly affects the comprehensiveness of the generated co-attention. In general, deeper convolution layers enable the fusion module with a larger receptive field, making each entry of the fused feature tensor aware of more context content and leading to more comprehensive attention cover on the whole object of interest (as shown in Figure 3.12). In other words, if more comprehensive spatial attention is desired, which can comprehensively cover the whole object of interest, enriching each local feature with more context information is necessary.

Chapter 4

Clustering based co-attention

4.1 Introduction

Although the CANet, proposed in the former chapter, could generate accurate co-attention maps even under challenging situations and lead to better retrieval accuracy, the extra computation cost caused by co-attention generation makes it impractical for large-scale image retrieval, as the computation costs, including both time cost and memory cost, are important performance metric of a CBIR framework. However, the good co-attention map output provided by CANet leads us to consider comparing the global feature extracted from the query image with each local feature extracted from a candidate image, resulting in the co-attention map generation. The central problem is to figure out a more efficient co-attention enabled CBIR framework that still keeps the advantage of co-attention but with acceptable extra computation cost.

This chapter proposes a more straightforward, efficient, and effective co-attention mechanism for large-scale image retrieval. Following the conclusion drawn from the CANet experiments, the proposed co-attention method also generates query sensitive co-attention maps in a global-to-local manner. Unlike the CANet proposed in the former chapter, which still requires separate attention network branch training, the co-attention method described in this chapter is based on feature tensor output by pre-trained CNN backbone networks without any extra trainable layer or network structure modification. In other words, the proposed co-attention method could be treated as a post-processing

module for convolution feature re-weighting conditioned on query content. To reduce the computation cost caused by query sensitivity, the L2 norm based feature selection and local feature clustering are applied, making the proposed co-attention practical even for large-scale image retrieval. According to experimental results, the proposed co-attention method could also generate good co-attention maps even for some really hard image retrieval cases. When embedding the generated co-attention method into the image feature extraction pipeline, image retrieval performance is significantly improved, leading to new state-of-the-art results on the benchmark evaluation datasets.

The rest of the chapter has the following content. Some insights into the generalized mean pooling and the baseline model structure are introduced in Section 4.2. The pipeline of the proposed co-attention method is illustrated in Section 4.3, followed by some extra processing steps for further computation cost reduction and online retrieval speeding up. Experimental results are demonstrated in Section 4.5. Moreover, Section 4.6 shows comprehensive ablation studies about the impact of different modules or hyper-parameter settings within the proposed co-attention method on the image retrieval performance. The final conclusions are drawn in Section 4.7.

4.2 Preliminary

The proposed co-attention mechanism could be treated as a post-processing module for feature re-weighting on the convolution feature tensor. In other words, it does not involve any training but is directly applied with the feature tensor output by pre-trained CNN-based spatial pooling CBIR model. Accordingly, in this section, we first discuss some insights about spatial pooling. Then, we introduce the structure and training details of the baseline model which serves as the feature extractor for the co-attention method.

4.2.1 Spatial pooling

Let us consider that an input image $\mathbf{I}$, after feeding through a convolution neural network, it is mapped into a feature tensor $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$, where H, W, D represent the height, width and channel counts. Global spatial pooling compresses the feature tensor $\mathbf{X} =$

$[x_{l,d}] \in \mathbb{R}^{L \times D}$ into a compact feature vector $\mathbf{V} = [v_d] \in \mathbb{R}^D$ using :

$$v_d = \left(\frac{1}{L} \sum_{l=1}^{L} x_{l,d}^p \right)^{\frac{1}{p}}, \tag{4.1}$$

where $L = H \times W$, $l = 1, \ldots, L$ and $x_{l,d}$ indicates the element from channel d of $\mathbf{X}$ at location l. p is a trainable power coefficient. Each element v_d of global spatial pooling feature vector $\mathbf{V}$ is a sum of feature maps at the channel d from the original feature tensor $\mathbf{X}$ raised to the power p. The ratio between each specific feature tensor element $x_{l,d}$ and the feature vector element v_d is expressed as:

$$\begin{aligned} r_{x_{l,d}} &= \frac{x_{l,d}}{v_d} \\ &= \frac{x_{l,d}}{\left(\frac{1}{L}\right)^{\frac{1}{p}} \left(\sum_{l'=1}^{L} {x_{l',d}}^p \right)^{\frac{1}{p}}} \\ &= L^{\frac{1}{p}} \left(\frac{{x_{l,d}}^p}{{x_{1,d}}^p + {x_{2,d}}^p + \ldots + {x_{l,d}}^p + \ldots + {x_{L,d}}^p} \right)^{\frac{1}{p}} \\ &= L^{\frac{1}{p}} \left(\frac{1}{(\frac{x_{1,d}}{x_{l,d}})^p + (\frac{x_{2,d}}{x_{l,d}})^p + \ldots + 1 + \ldots + (\frac{x_{L,d}}{x_{l,d}})^p} \right)^{\frac{1}{p}}. \end{aligned} \tag{4.2}$$

When $p = 1$, v_d is the mean of each feature map element $x_{l,d}$ at channel d, and the pooling result equals to the global average pooling (sum-pooling) [7]. When $p \to \infty$, according to Eq. (4.2), $r_{x_{max,d}} \to 1$ ($x_{\max,d} = \max_l x_{l,d}$), and it has $v_d \to x_{\max,d}$, and the pooling gives similar result to the max-pooling [124]. When $p \in (1, \infty)$ it is the so called Generalized Mean pooling (GeM) [98]. Thus, the sum-pooling and the max-pooling could be treated as special cases of the GeM. This also explains why GeM outperforms the other two pooling methods. Due to the usage of the power coefficient p, the GeM is more selective than simple sum-pooing while involving more local feature information than the max-pooling into the estimation process, leading to more comprehensive feature extraction.

Normally, the similarity measure between global spatial pooling feature vectors is performed using cosine similarity or L2 distance (after being L2-normalized). Considering the query image I_q, candidate image I_c along with corresponding feature tensors $\mathbf{X}_q$, $\mathbf{X}_c$

and global spatial pooling feature vectors $\mathbf{V}_q$ and $\mathbf{V}_c$, their cosine similarity is given by:

$$\begin{aligned} cos(\mathbf{V}_q, \mathbf{V}_c) &= (\eta(\mathbf{V}_q)\mathbf{V}_q)(\eta(\mathbf{V}_c)\mathbf{V}_c)^\mathsf{T} \\ &= \eta(\mathbf{V}_q)\eta(\mathbf{V}_c)\sum_{d=1}^{D} v_{q,d}v_{c,d} \\ &= \frac{\eta(\mathbf{V}_q)\eta(\mathbf{V}_c)}{(L_qL_c)^{\frac{1}{p}}}\sum_{d=1}^{D}\left(\sum_{l_q=1}^{L_q}\sum_{l_c=1}^{L_c}\left(x_{q,l_q,d}x_{c,l_c,d}\right)^p\right)^{\frac{1}{p}} \end{aligned} \tag{4.3}$$

where L2 normalization is defined by $\eta(\mathbf{V}) = \mathbf{1}/\left\|\mathbf{V}\right\|$. As claimed by Eq. (4.3), the cosine similarity between two global spatial pooling feature vectors can be treated as the sum of dimension-wise multiplications between the entries of the feature tensors, which represent the query image and candidate image separately.

According to [123], at the training stage, any loss function, such as the contrastive loss [24] or the triplet loss [4], that tries to optimize the cosine similarity between global spatial pooling feature vectors, would implicitly optimize the following aspects: first, the content from locations that contain background characterized by uniformly consistent information, such as sky, sand, and grass, is usually shared among many images. They are not distinctive and could not be utilized to distinguish two distinct images or to find correspondences between two matching ones. Accordingly, the activation value across all channels when considering such plain background locations tends to be zero ($x_{l_{bg},d} \to 0$), leading to little or no contribution to the final similarity score. On the contrary, locations of distinct foreground objects or regions tend to have large absolute values across all channels ($\left|x_{l_{fg},d}\right|$ is maximized), resulting in significant contributions to the final similarity score. Meanwhile, for foreground location pairs, which depict the matching objects or regions between $\mathbf{I}_q$ and $\mathbf{I}_c$, their feature representations are pushed closer together such that it yields a large positive product value. Conversely, for the location pairs that depict unmatching objects, their feature representations are pushed away from each other, yielding negative values for the final similarity score in Eq. (4.3).

The cosine similarity between spatial pooling feature vectors from Eq. (4.3) provides a useful hint to the CNN model training: optimizing the global spatial pooling feature vector's cosine similarity between image pairs implicitly optimizes the local feature matching. Foreground locations on the feature tensor $\mathbf{X}$ would be activated with high absolute fea-

ture values across all channels, resulting in large L2 norms (as well as L1 norms), while the background locations would have low feature activation values. Accordingly, the L2 norm of each entry on the feature tensor could be treated as spatial attention that the spatial pooling model implicitly learns at the training stage.

4.2.2 Baseline model structure and training

The general framework of using a deep CNN for feature tensor extraction followed by a global spatial pooling layer for building a compact global feature vector has been used in recent state-of-the-art works, such as DELG [14] and DOLG [146]. In this chapter, we also use ResNet [45] as the backbone network for feature tensor extraction. The feature tensor output by the final convolution layer is pooled by a generalized mean pooling layer from Eq. (4.1), with a fixed power co-efficient $p = 3$, followed by a trainable fully connected layer for feature whitening.

Following the approach in DELG [14], we also consider image-level class labels and the ArcFace margin loss [29] for the model training, defined by:

$$L(\widehat{\mathbf{V}}_g, \mathbf{y}) = -\log\left(\frac{\exp(\gamma \times \mathrm{AF}(\widehat{\mathbf{V}}_g \widehat{\mathbf{w}}_i^{\mathsf{T}}, y_i))}{\sum_{j=1}^{N_c} \exp(\gamma \times \mathrm{AF}(\widehat{\mathbf{V}}_g \widehat{\mathbf{w}}_j^{\mathsf{T}}, y_j))}\right), \tag{4.4}$$

where $\widehat{\mathbf{V}}_g$ is the whitened L2 normalized global GeM feature vector for each input training image, $\widehat{\mathbf{w}}_i$ refers to the trainable L2 normalized classifier weights for class i from the ArcFace weight matrix $\mathcal{W} \in \mathbb{R}^{N_c \times D}$. N_c is the number of classes in the training dataset, $\mathbf{y}$ is a one-hot class label vector and i is the index of the ground-truth class of $\widehat{\mathbf{V}}_g$ ($y_i = 1$) and γ is a trainable temperature parameter. $\mathrm{AF}(u, y)$ is the ArcFace-adjusted cosine similarity [14]:

$$\mathrm{AF}(u, y) = \begin{cases} \cos(\arccos(u) + m), & \text{if } y = 1 \\ u, & \text{if } y = 0 \end{cases} \tag{4.5}$$

where u is the cosine similarity, y indicates whether it is the ground-truth class and m is the ArcFace margin.

The ArcFace margin loss from Eq. (4.5) could also be referred as a "cosine classifier" [14]. Within the ArcFace weight matrix $\mathcal{W}$, each row $\mathbf{w}_i \in \mathbb{R}^{1\times D}$, $i \in \{1, 2, 3, \ldots, N_c\}$ can be treated as a proxy feature vector for class i. In other words, each proxy feature models representative information of each class and the ArcFace loss potentially optimizes the cosine similarity not between single image pairs but between each training image and proxies of classes. Compared to the traditional image pair similarity loss (contrastive loss or triplet loss) this kind of proxy-based similarity loss does not need hard sample mining and would converge faster than the simple similarity loss between specific image pairs [82].

4.3 Enabling CBIR with co-attention

In the following, we consider using the convolution feature tensor output by the pre-trained CNN model for enabling the co-attention generation process. The baseline GeM model, which is trained as described in Section 4.2.2, is used for feature extraction without considering any parameter fine-tuning or structure modification.

4.3.1 A naive way for co-attention generation

Let us consider a pair of images, representing the query image $\mathbf{I}_q$ and the candidate image $\mathbf{I}_c$ from a given database. After feeding through the backbone CNN, these images yield the feature tensors $\mathbf{X}_q \in \mathbb{R}^{H_q\times W_q\times D}$ and $\mathbf{X}_c \in \mathbb{R}^{H_c\times W_c\times D}$ as the outputs. The former query tensor is transformed into a compact query feature vector $\mathbf{V}_q \in \mathbb{R}^D$ by the spatial pooling using Eq. (4.1). The latter feature tensor $\mathbf{X}_c$, resulting from the final convolutional layer, models the grid-structured representations according to the corresponding locations for the candidate image. The precision of the correspondence between each entry on the feature tensor and regions on the input image depends on the processing properties of the CNN backbone structure. For example, ResNet [45] contains 5 blocks, each down-sampling the input feature tensor by half. After feeding through it, each local feature from the output feature tensor $\mathbf{X}_c$ corresponds to a 32×32 ($2^5 = 32$) pixels region from the input image.

A naive and straightforward way to get the co-attention map $\mathbf{a}_{naive} = [a_{l_c}] \in \mathbb{R}^{H_c \times W_c}$ of candidate image $\mathbf{I}_c$ with respect to the query image $\mathbf{I}_q$ could be simply calculating the cosine similarity between the global query feature vector $\mathbf{V_q}$ and the candidate feature tensor $\mathbf{X}_c$ from each location, as

$$a_{l_c} = \widehat{\mathbf{V}}_q \widehat{\mathbf{x}}_{c,l_c}^{\mathsf{T}}, \tag{4.6}$$

where $\widehat{\mathbf{V}}_q$ represents the whitened (by the pre-trained fully connected layer) and L2 normalized query feature $\mathbf{V}_q$. $\widehat{\mathbf{x}}_{c,l_c} \in \mathbb{R}^D$ is a local feature vector at location l_c from the candidate image feature tensor $\mathbf{X}_c$ that has been whitened and then L2 normalized. Softmax operation is applied on $a_{l_c} \in [-1, 1]$ to normalize their values into the range $[0, 1]$:

$$a'_{l_c} = \frac{\exp(a_{l_c})}{\sum_{i=1}^{K} \exp(a_i)} \ . \tag{4.7}$$

The visualization comparison between the L2 norm attention and the naive co-attention is provided in Figure 4.1. The L2 norm attention maps, shown in the third column of Figure 4.1, are obtained by calculating the L2 norm for each location on the feature tensor $\mathbf{X}_c$. The resulting attention map is then resized to the original image size and overlapped on the image as a heat-map. We can observe that L2 norm attention maps tend to highlight representative parts of all landmark buildings. The naive co-attention maps, shown in the fourth column of Figure 4.1, are visualizations of the results provided by Eq. (4.6) and Eq. (4.7). We can observe that simple cosine similarity between candidate local features and query global features could already give some generally good co-attention results. The first row from Figure 4.1 shows an easy case of image retrieval, in which the target object is salient and rather large scale in the candidate image without any distractors around, both L2 norm attention and the naive co-attention show corresponding reasonable highlight regions. For the hard case from the second row of Figure 4.1, the target object is not only small and remote but there are some similar class building architectures nearby, the naive co-attention highlights the correct region while the L2 norm highlights many irrelevant distractor objects and regions.

Although the discussions mentioned above demonstrate the validity of co-attention that is generated based on global-to-local feature match using cosine similarity, there are still two main problems with this naive implementation of co-attention. First, although the

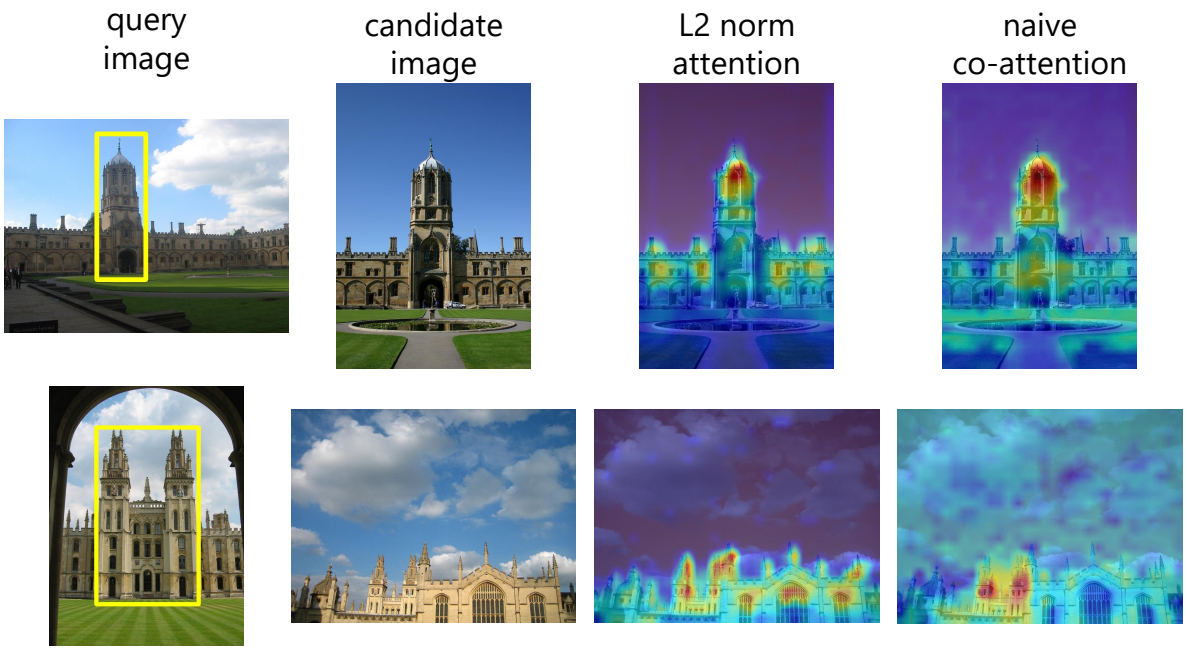


Figure 4.1: Visualization comparison of L2 norm attention and the naive co-attention. The first column shows the query images with a yellow bounding box outlining the target object. The second column shows the candidate image. The third column shows the L2 norm attention while the fourth column represents the result of the naive co-attention as described in Section 4.3.1.

deep structure of deep CNN enables them with large receptive fields, they may still not be comprehensive enough, as each local feature only corresponds to a grid local region on the original image. For example, it may only correspond to a small part of the target object and miss the high-level semantic meaning, which could result in some unwanted highlight regions or even noisy undefined regions.

Second, but also the most critical problem with the method described above is its computation cost. Consider an input image $\mathbf{I}$ of size $h \times w$, after feeding through ResNet [45], the output feature tensor $\mathbf{X}$ is of size $\frac{h}{2^5} \times \frac{w}{2^5}$. For a high-resolution image, such of 1024×1024 pixels, the output candidate feature tensor size could be as large as 32×32. For each element of these local features, if we have a 4 Byte float number for representation, the total memory cost for each candidate image local feature caching will be $2048 \times (\frac{1024}{32})^2 \times 4$ Bytes ≈ 8 MB, where 2048 is the channel count for the feature output by ResNet. If considering the multi-scale feature extraction scheme [98], the memory cost would increase exponentially. Pre-caching that many local features for a large image retrieval database is impractical.

In the following, we aim to make the co-attention mechanism described in Section 4.3.1

efficient and practical even for large-scale image retrieval task.

4.3.2 Co-attention enabled through feature selection and clustering

Local feature selection and clustering. As mentioned above, the most critical problem for using co-attention is the computation cost caused by the large number of local features that could possibly be extracted from a single image. An intuitive way to reduce the extra cost is to decrease the number of local features kept for each database image. Not all these local features from the feature tensor are relevant for CBIR tasks. Many irrelevant local features, such as those corresponding to the background, should be discarded from further processing, ensuring robustness and computational efficiency. Accordingly, we first perform local feature selection over the feature tensor output by the backbone network. As discussed in Section 4.2.1, the L2 norm of each entry from the CNN feature tensor can reflect its importance. With the input image $\mathbf{I}$ as input, the feature selection is performed based on the L2 norm attention of its feature tensor $\mathbf{X}$. We keep the top N features with the highest L2 norm, resulting in a set of local features $\mathbf{X}_N = [\mathbf{x}_n] \in \mathbb{R}^{N \times D}$, where $n = 1, \ldots, N$ and $\mathbf{x}_n \in \mathbb{R}^{1 \times D}$ indicates the n-th local feature vector from the set $\mathbf{X}_N$.

At this stage, each local feature can be treated as corresponding to a localized region from the input image. As mentioned before, these localized features may not be comprehensive enough to represent the whole object or regions of interest. Meanwhile, we want to further reduce the number of candidate local features for the sake of controlling the computational complexity at the online retrieval stage. Clustering is well known as an unsupervised approach for data reduction and we employ k-means clustering in order to extract fewer but more representative features from $\mathbf{X}_N$. However, the clustering result for k-means could vary with the cluster center initialization. This is an unwanted attribute for a stable image retrieval system, so we adapt the k-means++ [5] for the cluster center initialization. Considering the candidate image local features $\mathbf{X}_N$ as input, we consider the following steps for k-means clustering initialization:

1. Let $\mathbf{O}$ represent the local features from $\mathbf{X}_N$ that have not been selected as initial

centers, while $\mathbf{Z}$ represents the chosen local features set. In the beginning, $\mathbf{O} = \{\mathbf{x}_i | i = 1, \ldots, N\}$ and $\mathbf{Z} = \emptyset$.

2. Choose $\mathbf{x}_m \in \mathbf{O}$, such that $m = \arg\max_{\mathbf{x}_i \in \mathbf{O}} \|\mathbf{x}_i\|$, as the first cluster center. Meanwhile, add $Z = Z \bigcup \mathbf{x}_m$, while this is deleted from $\mathbf{O} = \mathbf{O} \setminus \mathbf{x}_m$.

3. For each local feature vector $\mathbf{x}_i \in \mathbf{O}$ that has not been chosen as a center yet, compute the smallest distance with respect to all chosen initial centers $d(\mathbf{x}_i) = \min \|\mathbf{x}_i - \mathbf{x}_j\|$, $\mathbf{x}_j \in \mathbf{Z}$, $j = 1, \ldots, |\mathbf{Z}|$, where $|\cdot|$ denotes the cardinality of a set.

4. Choose $\mathbf{x}_l \in \mathbf{O}$, such that $l = \arg\max_{\mathbf{x}_i \in \mathbf{O}} d(\mathbf{x}_i)$ as another cluster center, adding it to $Z = Z \bigcup \mathbf{x}_l$ and deleting it from $\mathbf{O} = \mathbf{O} \setminus \mathbf{x}_l$.

5. Repeat Steps 3) and 4) until $|\mathbf{Z}| \equiv K$.

The selected cluster centres are then used to initialise the standard k-means clustering. After clustering, we perform generalized mean pooling as in Eq. (4.1) within each cluster followed by whitening to obtain a set of clustered local features $\mathbf{X}_K \in \mathbb{R}^{K \times D}$, where $k = 1, \ldots, K$, for the input image.

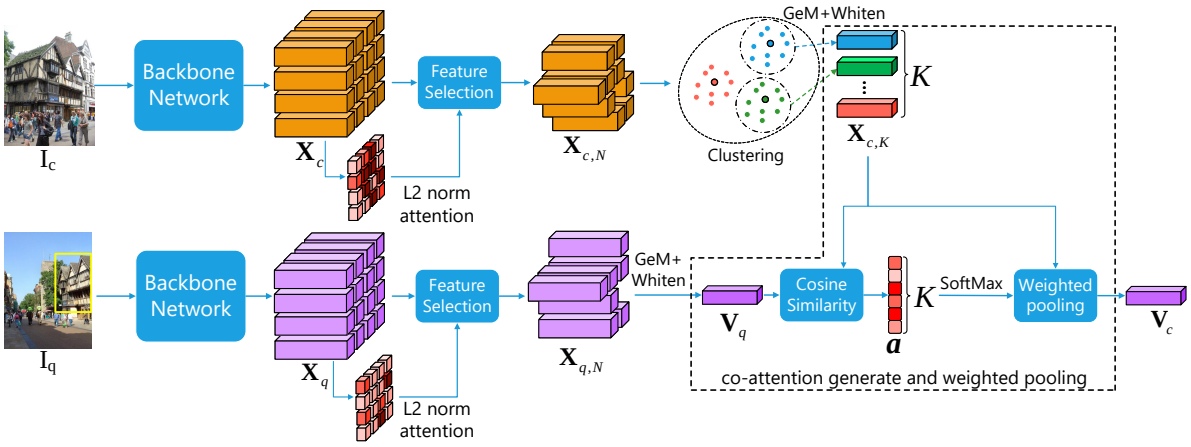


Figure 4.2: Illustration of clustering based co-attention generation and weighted feature extraction.

Co-attention generation with clustered local features The pipeline of co-attention generation and weighted feature extraction is illustrated in Figure. 4.2. Still consider the pair of images, representing the query image $\mathbf{I_q}$ and the candidate image $\mathbf{I_c}$. After feeding through the backbone network, followed by the local feature selection based on the L2 norm, selected query local features $\mathbf{X}_{q,N}$ are directly GeM pooled and whitened to obtain the query global feature $\mathbf{V}_q$. Selected candidate local features $\mathbf{X}_{c,N}$ are clustered and

then whitened as in Figure 4.2, resulting in the clustered local feature set $\mathbf{X}_{c,K}$. Then, the co-attention weights $\mathbf{a} = [a_i] \in \mathbb{R}^K$ are obtained by calculating the cosine similarity between $\mathbf{V}_q$ and each local feature from $\mathbf{X}_{c,K}$. As the attention weights are calculated by cosine similarity between the query and candidate image features, its range is between $[-1, 1]$ which may not ensure a high contrast among the results. To normalize values into the range $[0, 1]$ and better control the weight distribution, a SoftMax function defined by a temperature parameter T is applied on $\mathbf{a}$:

$$a'_i = \frac{\exp(a_i T)}{\sum_j^K \exp(a_j T)} \tag{4.8}$$

The final co-attention weighted candidate image global feature vector $\mathbf{V}_c$ is defined by weighted sum pooling:

$$\mathbf{V}_c = \frac{1}{K} \sum_i^K a_i \mathbf{X}_{c,i}. \tag{4.9}$$

The similarity measure is performed by evaluating the cosine similarity between $\mathbf{V}_q$ and $\mathbf{V}_c$.

4.4 Further computation cost reduction

In this section, to make the proposed co-attention more practical for large-scale image retrieval and to further reduce the computation cost we propose two extra processing steps during the retrieval stage.

4.4.1 Dimension reduction by PCA

Principal component analysis (PCA) has been used as a common method for feature dimension reduction. Unlike some other works that jointly perform dimension reduction and feature whitening by one fully connected layer [123], we perform dimension reduction by employing Principal Component Analysis (PCA) as a post-processing step. There are two main reasons to use the PCA: first, we found that training with the original feature dimension (2048 for ResNet) makes the model converge faster; second, it is more

convenient and fair to compare retrieval performance with different dimension settings as all experiments are based on one same pre-trained model. For the query image, PCA dimension reduction is applied on its whitened global feature vector $\mathbf{V}_q$. For the candidate image local features, PCA is applied on each whitened local feature from $\mathbf{X}_{c,K}$ before L2 normalization.

In our implementation, PCA parameters: mean and eigenvectors, which are denoted as $\mathbf{m}_v \in \mathbb{R}^{1\times D}$ and $\mathbf{P}_v \in \mathbb{R}^{D^{'}\times D}$ and used for dimension reduction of $\mathbf{V}_q$ and $\mathbf{X}_{c,K}$, are learned from whitened global GeM pooling feature vectors (without L2 normalization) of random images from the training dataset. D equals to the original feature dimension output by the backbone network while $D^{'}$ denotes the feature dimension after PCA dimension reduction.

4.4.2 Speed up retrieval with inverted file indexing

For image retrieval, especially on a large-scale candidate image database, it may not necessary to apply co-attention for each candidate image feature extraction. Actually, some candidate images are quite distinguishable and they are not worth performing careful similarity measures with the query. To reduce the candidate image count that needs to be compared with the query image at the online retrieval stage, inverted file indexing [112] is added to the proposed co-attention method pipeline. This technique has been applied in previous work. For example, HOW [123] only performs feature comparison between the local features that share the same visual word. Similarly, after dimension reduction, we use local features from the feature tensor output by the final convolution layer to train the codebook. At the feature extraction stage, both query image and candidate image local features $\mathbf{X}_{c,N}$ and $\mathbf{X}_{q,N}$, after dimension reduction and whitening, are clustered over visual words from the codebook and we record the visual word indices that each image is assigned to. Then, at the retrieval stage, for each query image, we only pick out those candidate database images that at least share one visual word with the query image to perform co-attention generation and assess their similarity. Other candidate images which are not selected are simply set to have zero similarity score with the query image.

The global picture of the proposed co-attention enabled CBIR framework when considering the inverted file indexing is shown in Figure. 4.3.

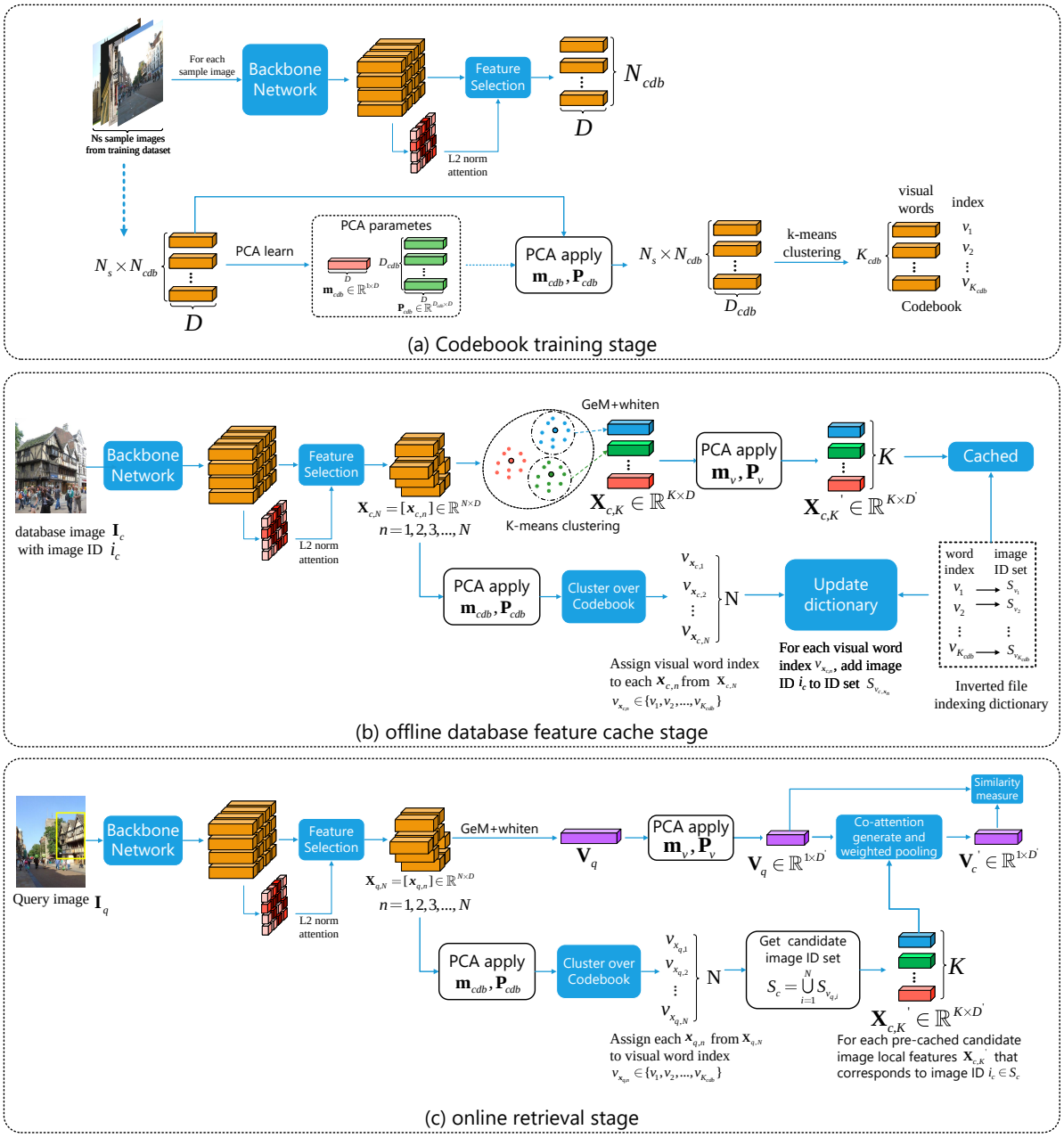


Figure 4.3: Illustration of our method pipeline with inverted file indexing.

Codebook training. The inverted file indexing starts with the codebook training. As shown in Figure 4.3 (a), at the codebook training stage, each sample image is fed through the pre-trained backbone network followed by the L2 norm based feature selection, resulting in N_{cdb} local features. With N_s sample images from the training dataset, there would be $N_s \times N_{cdb}$ sample local features. To reduce the computation cost, PCA dimension reduction is applied with parameters $\mathbf{m}_{cdb} \in \mathbb{R}^{1\times D}$ and $\mathbf{P}_{cdb} \in \mathbb{R}^{D_{cdb}\times D}$, which are learned from these local features. After the PCA dimension reduction, k-means clustering is employed to get the final K_{cdb} visual words, of indexes $\{v_1, v_2, \ldots, v_{K_{cdb}}\}$, as the codebook.

Feature caching. As shown in Figure 4.3 (b), during the offline database image feature extraction and caching stage, each database image $\mathbf{I}_c$ is fed through the backbone network. After the feature selection, with the output selected local features $\mathbf{X}_{c,N}$, one branch performs k-means clustering followed by PCA dimension reduction with parameters $\mathbf{m}_v$ and $\mathbf{P}_v$, resulting in the dimension reduced clustered local features $\mathbf{X}_{c,K}{}' \in \mathbb{R}^{1\times D'}$. Another branch, after PCA dimension reduction with parameters $\mathbf{m}_{cdb}$ and $\mathbf{P}_{cdb}$, $\mathbf{X}_{c,N}$ are clustered over the codebook, assigning each local feature to the closest visual word. A dictionary is used to record the database image ID for each visual word index. Each key of this dictionary is a visual word index. Each visual word index corresponds to a set of database image IDs whose local features are assigned to this visual word. The dictionary is updated when considering each database (candidate) image as input.

Online retrieval. As shown in Figure 4.3 (c), at the online retrieval stage, the selected query image local features $\mathbf{X}_{q,N}$, after PCA dimension reduction with $\mathbf{m}_{cdb}$ and $\mathbf{P}_{cdb}$, are also clustered over the codebook. Then, based on the cached dictionary, we only pick out those database images that would share at least one visual word with the query image for later co-attention weighted feature extraction (as shown in Figure. 4.2) and similarity measure. All other candidate images are treated as having a 0 similarity score to this query and removed from the image search. Remember that with inverted file indexing, the only extra thing that needed to be cached is the visual word index dictionary and the codebook, so it would hardly require extra memory.

4.4.3 Multi-scale feature extraction scheme

The size of the objects in the image could change dramatically when changing the image acquisition parameters. For example, when shooting images under various perspective projection conditions, objects may be located at various distances from the image plane and they would appear larger or smaller under different views and with various distortions. To solve identifying the scene from images taken under various acquisition conditions, at the retrieval stage, following the common practice from [98], we implement the multi-scale image feature extraction scheme. The multi-scale scheme is performed by resizing the input image with several scale factors and then considering the images of all scales. For the baseline GeM model without co-attention, the global feature vectors from different

scales are fused by average-pooling and then L2 normalized again as described in [98]. In our proposed co-attention enabled pipeline, local features from all scales are merged and selected jointly according to their corresponding L2 norm attention scores.

4.5 Experiments

We initially discuss the experiment setup, including the hyper-parameter setting and implementation details. Then, we provide visualizations of co-attention generation results and the retrieval results of the proposed co-attention methods along with comparisons to existing state-of-the-art works.

4.5.1 Experiment setup

Implementation details. We consider ResNet101 (and ResNet50) [45] as the backbone network ($D = 2048$). For the baseline GeM model training, we set the margin $m = 0.15$ and temperature $\gamma = 30$ for the ArcFace loss in Eq. (4.4), respectively. We train the model on a clean subset of Google landmark dataset version 2 (GLDv2) [136], which contains more than 1.5M images grouped in 81,313 classes. GLDv2 was also used for training the state-of-the-art DELG [14] and DOLG [146] models. We consider data augmentation by randomly cropping, ratio distorting and then resizing images to 512×512 pixels. The model is optimized using the SGD optimizer with an initial learning rate of 0.05, weight decay of 0.0001, and batch size of 128 images. A cosine learning rate decay strategy is applied. The generalized mean pooling power coefficient from Eq. (4.1) is fixed as $p = 3$. The baseline model is trained with 4 NVIDIA Tesla GPUs and the model is trained for 50 epochs.

At the retrieval stage, for the co-attention mechanism described in Section 4.3.2, if not otherwise specified, we set the local feature selection count $N = 500$ and the number of clusters as $K = 10$ for k-means clustering and $T = 10$ for the SoftMax temperature in Eq. (4.8).

For dimension reduction of query image and candidate image features, the PCA compo-

nents $\mathbf{m}_v$ and $\mathbf{P}_v$ are learned with whitened global GeM pooling feature vectors (without L2 normalization) of 50,000 random images from the training dataset. After whitening, the global query image feature vector $\mathbf{V}_q$ and clustered candidate image local features $\mathbf{X}_{c,K}$ are compressed using the PCA dimension reduction with parameters $\mathbf{m}_v$ and $\mathbf{P}_v$ to dimenstion $D^{'} = 512$.

For the inverted file indexing, we use $N_s = 60,000$ random images in a single original scale from the training dataset (GLDv2), with $N_{cdb} = 300$ local features being selected from each of them to train the codebook. The size (cluster count) of codebook $K_{cdb} = 65536$. For computation cost reduction, PCA parameters $\mathbf{m}_{cdb}$ and $\mathbf{P}_{cdb}$ are learned from these sample features and used to compress them to dimension $D_{cdb} = 128$.

For the implementation of the multi-scale scheme feature extraction, as described in Section 4.4.3, when not considering co-attention, we use 3 scales $\{1, \sqrt{2}, \frac{1}{\sqrt{2}}\}$ as in other global feature approaches [98, 14]. When considering the co-attention, we have feature selection from 5 scales: $\{\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$, as in [121].

Evaluation datasets. Revisited Oxford and Paris datasets [96] have commonly been used for large-scale CBIR performance evaluation in recent years. These databases are expanded versions of Oxford [93] and Paris [94] datasets after removing the images with incorrect annotations and adding several new query images. Revisited Oxford (ROxf) contains 4993 images while Revisited Paris (RPar) has 6322 images. Both datasets contain 70 query images. The ground-truth matching images to each query image are divided into 3 groups, *Easy, Medium, Hard*, according to the level of difficulty in assessing the similarity of their image representation with the corresponding query. In addition, R1M [96] is a new distractor set containing 1 million unbiased high resolution (1024×768 pixels) images for ROxf and RPar. All retrieval results are reported with mean average precision (mAP), [93]. It should be mentioned that ROxf/RPar datasets provide bounding boxes for each query image, outlining the query object region. Meanwhile, the standard evaluation protocol requires cropping the query image with the bounding box as input.

4.5.2 Visualization of feature selection and clustering

We exemplify the results provided by the L2 norm based feature selection and k-means clustering on a set of images showing architectural landmarks in Figure 4.4. We can observe that the selected features are mainly distributed in the regions of the main landmarks displaying architectural details. The most representative parts of the buildings, like the two towers on the top of the building in the top row of images from Figure 4.4, have relatively higher attention scores. When applying k-means clustering, it tends to group the locations which are visually similar to each other while different parts of the building are assigned to different clusters through k-means clustering. The positions assigned to the same cluster, marked with the same color in the third column of images from Figure 4.4, are considered to share the corresponding clustered local feature from $\mathbf{X}_{c,K}$, as their representation. These examples show that the latent space clustering leads to dividing and grouping the local image features into fewer, but more comprehensively representative feature vectors, which are used for co-attention generation later.

4.5.3 Visualization of co-attention

Examples of co-attention generation, considering the baseline GeM model trained on the GLDv2 dataset with local feature selection and clustering, are shown in Figure. 4.5. The first and second columns show the query and target images. For the third column co-attention map, local features that are grouped into the same cluster share the corresponding clustered local feature as their representation. Co-attention scores for the locations that are not selected are set to zero. Co-attention scores of all local features from different input image scales are projected back to the corresponding regions on the original image and accumulated to get the final co-attention map. The L2 norm attention of the baseline GeM model is also visualized in the fourth column of Figure 4.5 for comparison. As discussed in Section 4.2.1, the L2 norm reflects the importance of each location with respect to how much it contributes to the final feature vector obtained by global pooling. In other words, the L2 norm is also a query non-sensitive attention that the spatial pooling model implicitly learned at the training stage.

In examples 1-4 from the top rows of Figure 4.5 some typical situations are shown, in

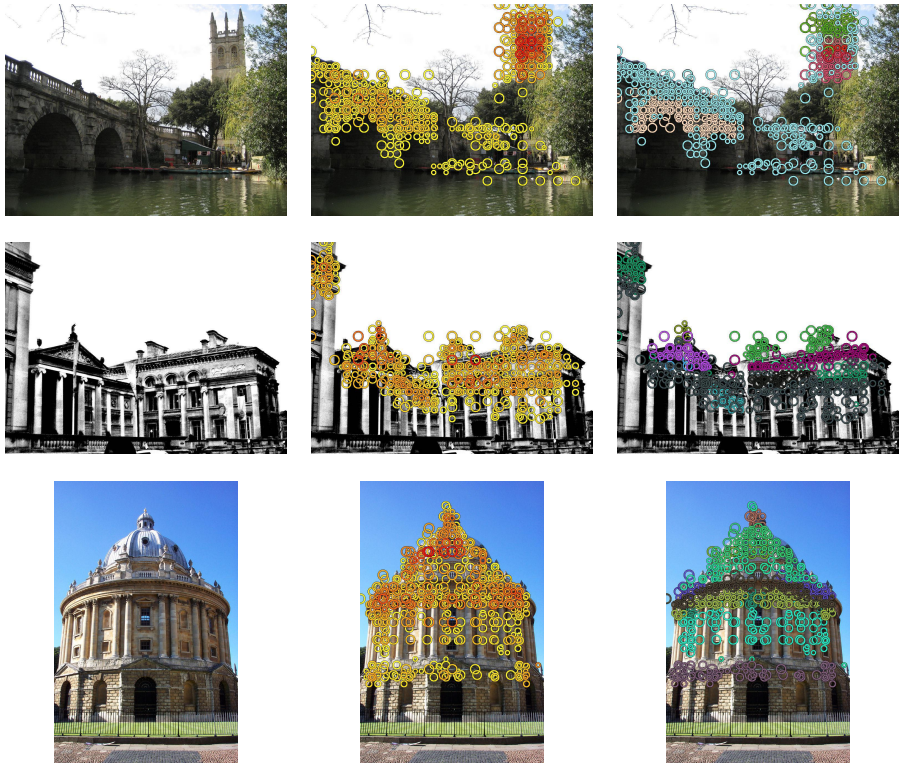


Figure 4.4: Visualization of the feature selection and k-means clustering for the proposed co-attention mechanism. The first column represents the original images, while on the images from the second column, we indicate the selected local features with circles. The radius size in the circles indicates the scale of the image where they originate. The colour variation for circles, from yellow to red, indicates an increasing L2 norm attention score, with red indicating the highest score. Finally, the third column of images shows the result of the k-means clustering over selected local features, where the local features assigned to the same cluster are marked with the same colour. In these examples, we consider $N = 500$ feature vectors and $K = 10$ clusters selected by the k-means clustering.

which the target object is not salient or there are similar distractors nearby. The L2 norm attention tends to uniformly highlight all potential relevant regions as it has no access to the actual query information, and its action is only driven by the knowledge learned during training. As a consequence, the L2 norm attention could successfully discard background regions, but it has no idea for which foreground object to look at. In example 4, the L2 norm attention almost ignores the desired query house from the remote part of the scene while wrongly laying most emphasis on the tower building which is more salient and appears as more significant. Example 5 shows another really hard example, in which the target building is not intact and only shows a small part of the resized tower in the top-left corner of the target image. Moreover, there is a spire at the right side of the target building, which is very similar to the top part of the query object. The L2 norm highlights mostly the area around that spire, while the proposed co-attention pays

attention to the window and edge structure for the correct target object. Examples 6 and 7 show the co-attention with the same target image but different query content. In example 6, when considering the whole building as a query, the dome region is central for co-attention generation. However, when using only a window as the query, the co-attention correctly focuses on the corresponding region on the target image, despite the dramatic change in the image acquisition conditions. These results indicate the high level of sensitivity of the proposed co-attention method to the query content.

For another set of cases, examples 8 and 9 from Figure 4.5 show some easy situations where the target object is salient enough and not surrounded by hard distractors. In this case, the co-attention mechanism and L2 norm both correctly highlight the target objects despite the challenges in the scene representations in these images due to illumination changes and the view perspective changes during image acquisition.

Examples 10-12 from the bottom three rows of images from Figure 4.5 show some cases when the proposed co-attention method fails or does not provide good enough results. In example 10, the co-attention pays more attention to regions outside the target object. Example 11 represents another very challenging case, in which there is not only a massive change in scaling but also the target skyway is blocked by the foreground gate structure, the proposed co-attention fails to accurately localize the target but just equally highlights some surrounding buildings. Example 12 is one of the hardest cases in which the query content is not even an intact building but a small sculpture attached as one of the architectural elements on the skyway between two historic buildings. In this case, the co-attention fails to highlight the target region accurately while it also pays attention to the surrounding regions.

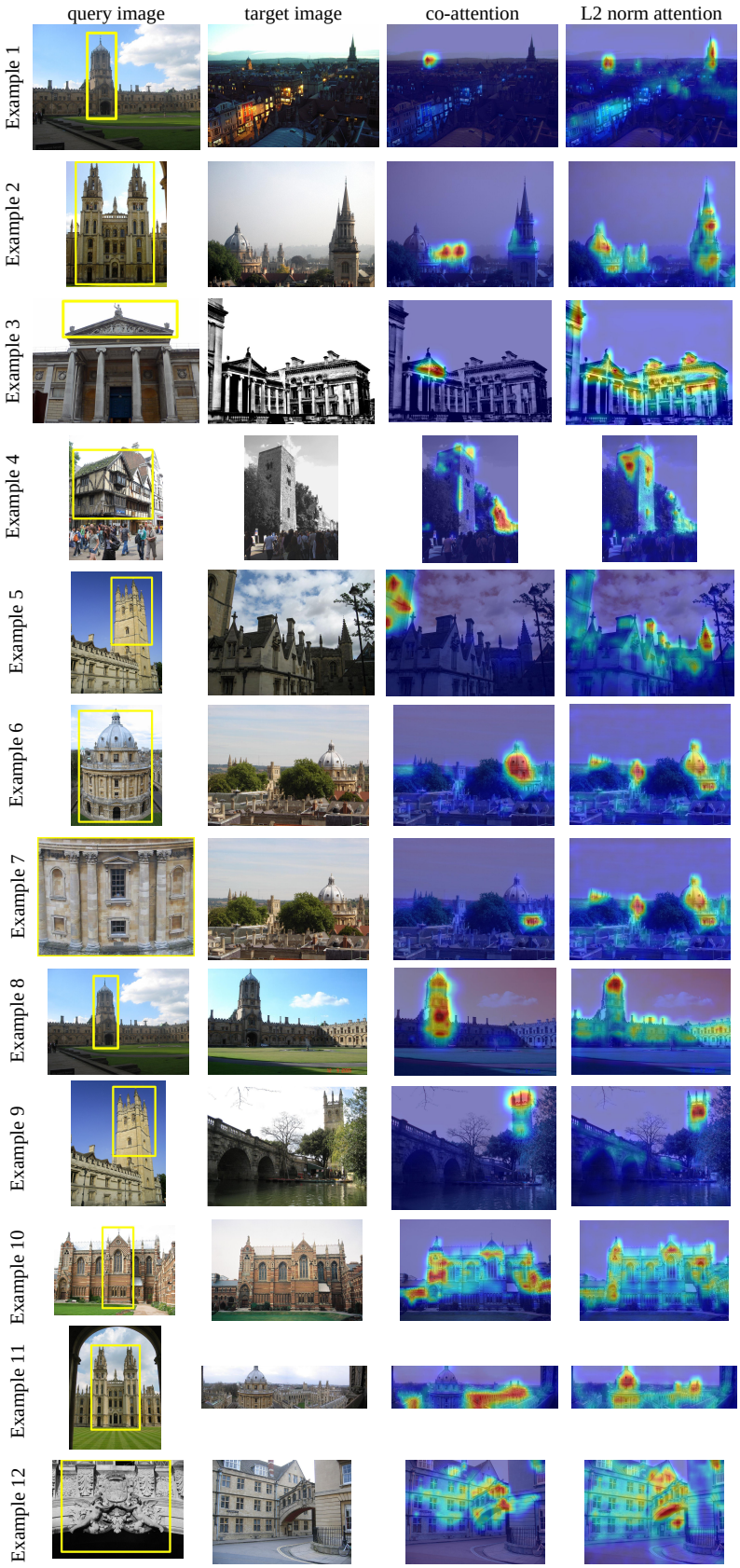


Figure 4.5: Attention map visualization. The first column shows the query image with a yellow bounding box outlining the target object. The second column is the target image. The third column represents the co-attention map while the final column is the L2 norm attention of the Generalized Mean pooling (GeM).

4.5.4 Image retrieval results

Image retrieval results for the proposed method and comparisons with other methods are provided in Table 4.1. As different existing CBIR works have different training settings, which may lead to unfair comparisons. We re-implement some of the best recent state-of-the-art (SOTA) methods according to the setting from Section 4.5.1 and marked with "†". In the following, we also focus on comparison with these SOTA works in Table 4.1.

Group (A) from Table 4.1 shows the results of local feature methods. The original SOTA work HOW is trained on rSfM-120k dataset [98] with contrastive loss [24]. We re-train it on the GLDv2 dataset with ResNet101 backbone and ArcFace loss, and it is indicated by R101$^-$-HOW (GLDv2)†[9]. Under our re-implementation, it greatly improved across all evaluation protocols, especially on ROxf *Hard* set, which reaches 71.3% mAP from 56.9% before. However, HOW has weak performance on RPar+1M dataset with the *Hard* evaluation protocol.

Group (B) from Table 4.1 shows the result of the global feature methods. They give worse results than the local feature method HOW on ROxf *Hard* set, but they show better generalization ability when considering the 1 million distractor set. The original DELG [14] was trained on GLDv2 with a small batch size of 32. We re-implement its ResNet101 version (R101-DELG†) under the training setting from Section 4.5.1. We point out that DELG first uses the global GeM feature for the initial retrieval result, then uses local features to perform spatial verification (SP) for re-ranking. The local feature branch of DELG model does not perform backward gradient transfer to its backbone network. In other words, without the second stage spatial verification (SP) re-ranking, the DELG model could be basically the same as a GeM model. We can see that the spatial verification results in a limited improvement, especially when considering the 1 million distractor set.

The bottom group (C) shows the results for the baseline model (GeM†) as described in Section 4.2.2 and when it is combined with the clustering-based co-attention method (GeM†-CA). In other words, for the results of GeM† and GeM†+CA, they share the same

[8]https://github.com/feymanpriv/DOLG

[9]R101$^-$ represents the ResNet101 without the final convolution block. According to the study from [123], HOW gives better results when discarding the final block, and we follow this setting for re-implementation.

Method	*Medium (%)*				*Hard (%)*			
	ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
(A) Local feature								
HesAff-rSIFT-ASMK*+SP [121]	60.6	46.8	61.4	42.3	36.7	26.9	35.0	16.8
HardNet-ASMK*+SP [79]	65.6	-	65.2	-	41.1	-	38.5	-
DELF-ASMK*+SP [120]	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
DELF-D2R-R-ASMK*+SP [120]	76.0	64.0	80.2	59.7	52.4	38.1	58.6	29.4
R50^{-}-HOW-MDA [137]	82.0	68.7	83.3	64.7	62.2	45.3	66.2	38.9
R50^{-}-HOW [123]	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
R101^{-}-HOW (GLDv2)†	83.9	77.9	87.9	76.4	71.3	52.8	76.0	56.4
(B) Global feature								
R101-R-MAC [39]	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
AlexNet-GeM [98]	43.3	24.2	58.0	29.9	17.1	9.4	29.7	8.4
VGG16-GeM [98]	61.9	42.6	69.3	45.4	33.7	19.0	44.3	19.1
R101-GeM [98]	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
R101-GeM-AP [101]	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
R101-GeM↑ [110]	67.3	49.5	80.6	57.3	44.3	25.7	61.5	29.8
R101-GeM (GLD) [85]	67.3	49.5	80.6	57.3	44.3	25.7	61.5	29.8
R101-DSM [110]	65.3	47.6	77.4	52.8	39.2	23.2	56.2	25.0
R101-SOLAR [85]	69.9	53.5	81.6	59.2	47.9	29.9	64.5	33.4
R50-DELG [14]	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
R50-DELG + SP [14]	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
R101-DELG [14]	76.3	63.7	86.6	70.6	55.6	37.5	72.4	46.9
R101-DELG + SP [14]	81.2	69.1	87.2	71.5	64.0	47.5	72.8	48.7
R101-DELG†	82.4	73.0	90.1	78.0	65.2	50.1	80.6	59.2
R101-DELG + SP†	84.1	75.9	91.0	79.2	68.8	53.6	83.0	62.3
R50-DOLG [146][8]	81.2	71.4	90.1	79.0	62.6	47.3	79.2	59.8
R101-DOLG [146][8]	82.3	73.6	90.9	80.4	64.9	51.6	81.7	62.9
(C) the proposed co-attention method								
R50-GeM†	79.8	69.0	87.3	73.1	60.4	44.2	74.0	52.0
R50-GeM†-CA	83.8	75.3	91.5	77.2	67.8	52.4	82.7	56.8
R101-GeM†	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
R101-GeM†-CA	**86.4**	**79.3**	**93.2**	**81.8**	**72.6**	**59.9**	**85.6**	**64.1**

Table 4.1: Image retrieval results on ROxf/RPar datasets and their extended versions when adding the 1 million distractor set R1M, for the *Medium* and *Hard* evaluation protocols. Groups (A) and (B) separately show the results of local and global feature methods, respectively. Group (C) shows the results of the proposed co-attention method. "†" indicates re-implemented model under the training details from Section 4.5.1. "SP" refers to the spatial verification re-ranking [87].

exact GeM backbone network with the training setting from Section 4.2.2, the only difference is that GeM†+CA implements the co-attention method as described in Section 4.3.2 (as well as PCA dimension reduction and inverted file indexing from Section 4.4) to re-weight the candidate image feature tensor before the global GeM pooling. It can be observed that introducing the co-attention to the CBIR pipeline can greatly improve the retrieval performance. Especially, on the *Hard* set of ROxf (RPar), GeM†+CA reaches the best result of 72.6% (85.6%). When considering the 1 million distractor set, the proposed

co-attention method still gives the best retrieval results.

4.5.5 Qualitative retrieval results

Figure 4.6 provides a qualitative comparison between the co-attention enabled GeM method "GeM†-CA" and the baseline retrained GeM model "GeM†", on the challenging ROxf dataset [96], considering the *Hard* evaluation protocol. The query image is shown on the first column from the left side of each row with a yellow bounding box indicating the query region of interest. The top 5 retrieval results are demonstrated, with the green outline denoting correct retrieval results while red markings denote incorrect results. The co-attention maps are shown below each row with the retrieved images. The proposed model globally outperforms the original GeM model, whose retrieved images are shown underneath. Especially in the top-left query example, the query region is not an intact building but only shows a structure from its middle part. GeM gives wrong results for three retrievals out of the top 5. Meanwhile, the co-attention method correctly provides all top 5 retrievals indicating the specific target region.

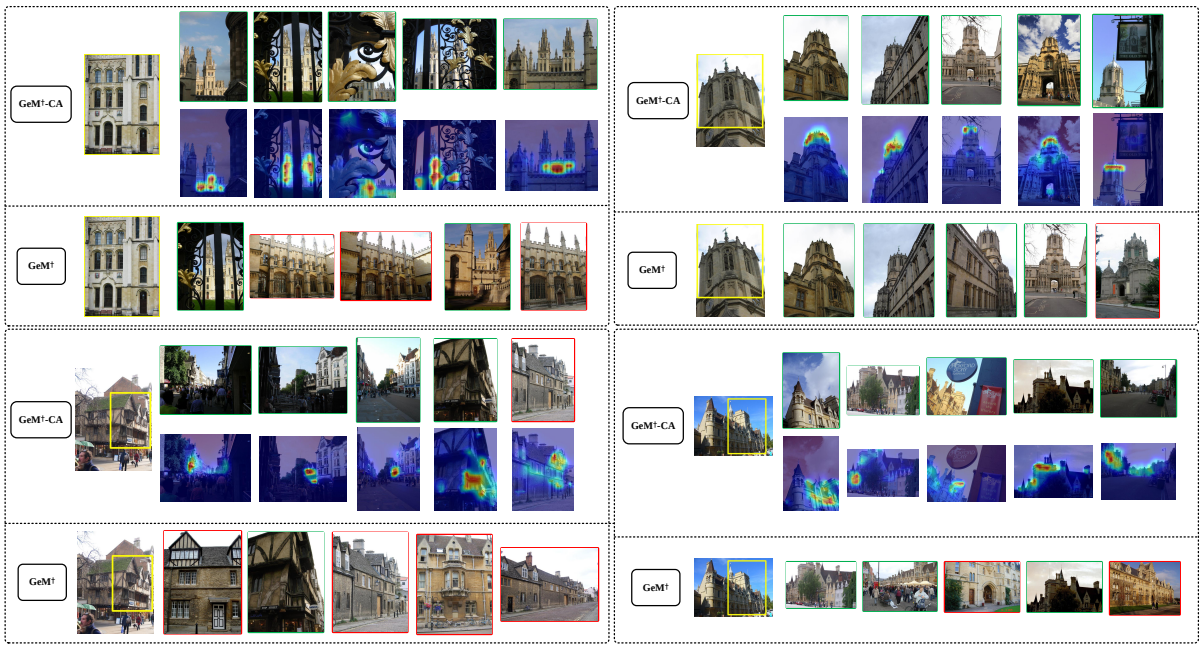


Figure 4.6: Top 5 retrieval results for GeM†-CA (with co-attention) and GeM† on images from hard set of ROxf dataset [96]. Co-attention maps are also provided underneath the retrievals provided by "GeM†-CA".

4.6 Ablation experiment and discussion

In this section, we present ablation experiment results for some hyper-parameter settings and discuss the computation cost of the proposed method.

4.6.1 Impact of local feature clustering

Apart from the computation cost reduction brought by clustering, we also test by implementing the co-attention without considering the clustering. It corresponds to the naive co-attention case described in Section 4.3.1. As shown in Table 4.2, the co-attention always improves the baseline GeM model's performance, even with the naive co-attention implementation. Although the proposed clustering procedure forcibly merges many local features into a few groups, which makes it lose some local feature information, it actually contributes positively to the final retrieval performance. This result again proves that considering clustering for co-attention not only relieves the extra computation cost caused by the query sensitivity but also further improves the retrieval results.

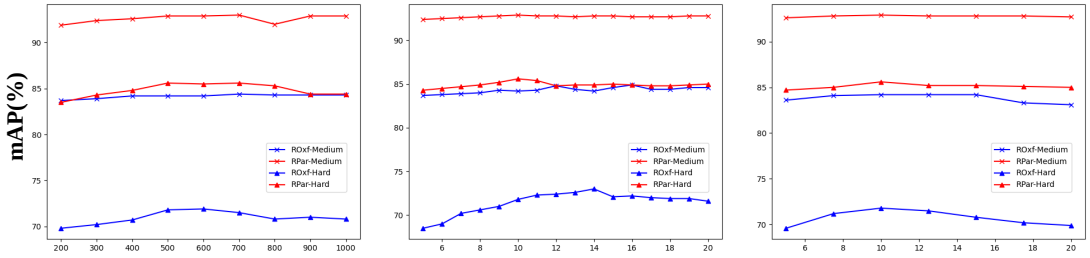
Model	co-attention	cluster	*Medium* (%)		*Hard* (%)	
			ROxf	RPar	ROxf	RPar
R101-GeM†	✗	✗	83.0	90.2	65.5	80.7
R101-GeM†-CA(naive)	✔	✗	83.7	90.4	69.9	80.9
R101-GeM†-CA	✔	✔	86.4	93.2	72.6	85.6

Table 4.2: CBIR mAP results on ROxf/RPar datasets with naive co-attention.

4.6.2 Impact of clustering parameters

Plots from Figures 4.7 (a), (b) and (c) show the impact of cluster hyper-parameters features N, clusters K, and the temperature T from Eq. (4.8), on the model retrieval performance. Generally, the proposed method is robust to changes in these hyper-parameters. The difference is mainly reflected in the ROxf *Hard* set. A small $N = 200$ could not cover enough local representative features, while a too large $N = 1000$ may pick out too many backgrounds or irrelevant local features, and also it will slow down the feature extraction procedure without bringing any obvious result improvement. Varying the number

of clusters K has implications not only on the performance but also on the computation cost. On the one hand, a smaller K could further reduce the computation cost but it will arbitrarily fuse many local features into larger clusters reducing the co-attention benefits. On the other hand, a larger number of clusters K could further improve the retrieval performance as it leads to smaller clusters. However, it will require additional computation costs and the improvement is minimal for $K > 16$.



(a) Ablation study on parameter N (b) Ablation study on parameter K (c) Ablation study on parameter T

Figure 4.7: Ablation experiment results when varying the clustering hyper-parameters.

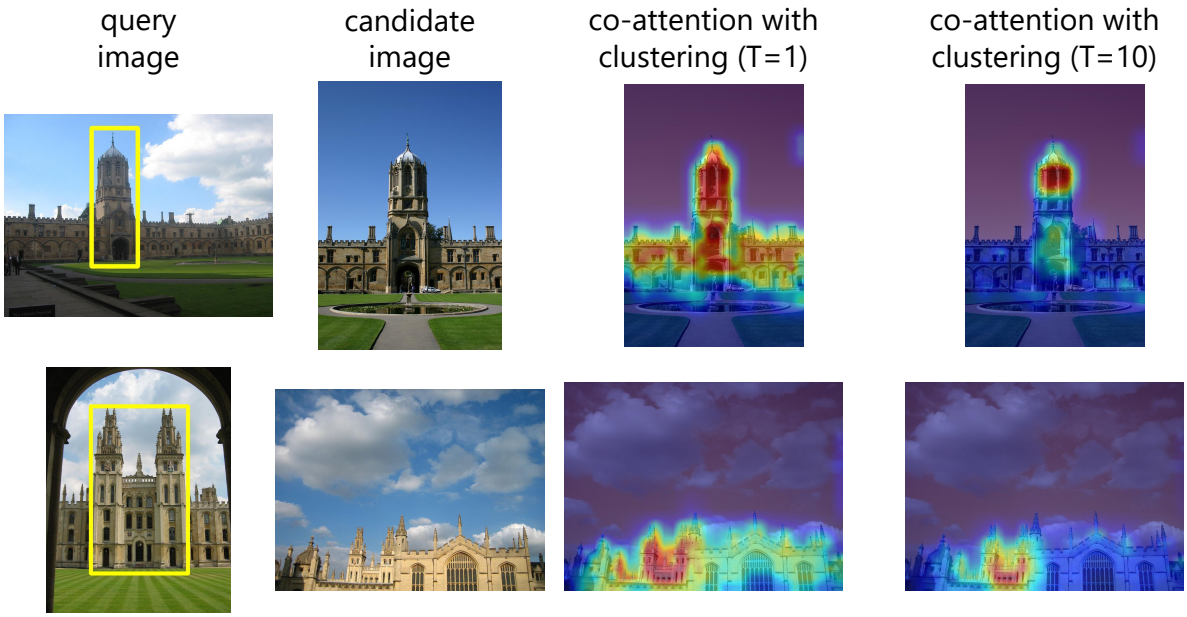


Figure 4.8: Co-attention map generated with clustering as described in Section 4.3.2, when T is set to 1 and 10.

To more clearly show the impact of parameter T on co-attention generation, some co-attention map, which are generated as described in Section 4.3.2, but with different T values for Eq. (4.8) are shown in Figure 4.8. As we can observe, for a small $T = 1$, the co-attention maps based on the clustered candidate image local features tend to cover more contextual regions of the target object. Nevertheless, there are still some unwanted regions. After considering a larger T = 10, the co-attention becomes much clearer and more focused on the target object.

4.6.3 Clustering method selection

This ablation study provides further discussion about the selection of the clustering method. In Section 4.2, modified k-means++ clustering is applied for the proposed co-attention method. Simple k-means clustering naturally has some disadvantages or limitations, such as it only works for convex shape distributed data and requires manually setting the number of cluster centres K. In the following, more tests are performed with two different clustering algorithms: Spectral Clustering [127] and Mean-Shift [22].

Spectral clustering is a graph-based clustering method that works well on some non-convex distributed data. In spectral clustering implementation, k-means is applied over the eigenvectors of the Laplacian of the graph and the cluster number is set to 10. As shown in Table 4.3, k-means++ and spectral clustering actually gives similar results. However, spectral clustering requires more computational requirements and consequently has a higher time cost. Additionally, another clustering method called Mean-Shift clustering is tested. Mean-Shift clustering is a kernel-based density estimation clustering that is very different from k-means clustering. The Mean Shift does not require the manual setting of the cluster centre count but requires setting a bandwidth parameter for the kernel. In Table 4.4 various bandwidth values are considered, such as 0.5, 1.0 and 1.5. Although by carefully setting the bandwidth value, the Mean-Shift could give a similar result to k-means, the number of features required is much larger than for k-means clustering. The number of local features is quite important as it directly influences the computation cost of the proposed co-attention method. According to these results, neither mean-shift nor spectral clustering are appropriate for implementing the co-attention mechanism.

Cluster Method	*Medium* (%)		*Hard* (%)	
	ROxf	RPar	ROxf	RPar
Spectral	86.4	93.1	72.7	85.6
k-means++	86.4	93.2	72.6	85.6

Table 4.3: Retrieval results on ROxf and RPar with Spectral Clustering.

Cluster method	Band width	Feature number	*Medium* (%)		*Hard* (%)	
			ROxf	RPar	ROxf	RPar
Mean-Shift	0.5	325	87.2	92.3	74.1	83.8
	1.0	76	85.4	91.6	71.7	82.6
	1.5	11	84.1	91.5	68.4	82.6

Table 4.4: Retrieval results on ROxf and RPar datasets with Mean-Shift clustering and different bandwidth setting. "feature number" indicates the average number of local features after clustering.

4.6.4 The impact of PCA dimension reduction

The experiment results for the feature vector reduction using PCA are shown in Table 4.5. According to Table 4.5, increasing the feature dimension from the default setting of 512 to 1024 will double the computation cost without bringing any significant improvement. On the contrary, considering a feature dimension of 256 or even smaller will lead to significant performance degradation. In conclusion, a feature dimension of 512 is a good balance between the performance and computation cost.

Feature Dimension	*Medium* (%)		*Hard* (%)	
	ROxf	RPar	ROxf	RPar
128	84.1	91.4	68.3	82.5
256	86.0	93.0	71.3	84.4
512	86.4	93.2	72.6	85.6
1024	86.4	93.2	72.7	85.7

Table 4.5: CBIR mAP results on ROxf and RPar datasets when varying the feature dimension.

4.6.5 Impact of scales

Retrieval results of the proposed method "GeM†+CA" with different image scales are provided in Table 4.6. There are different 3 existing scale combinations implemented in the literature : $\left\{\frac{1}{\sqrt{2}}, 1, \sqrt{2}\right\}$ from [98], $\left\{\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}\right\}$ from [146], and $\left\{\frac{1}{4}, \frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}, 2\right\}$ from [14, 123]. According to Table 4.6, when considering the combination of 5 scales gives the best result for the proposed co-attention method. Using 7 scales does not bring much improvement while increasing the computational cost for the feature extraction.

1	$\frac{1}{\sqrt{2}}$	$\sqrt{2}$	$\frac{1}{2\sqrt{2}}$	$\frac{1}{2}$	$\frac{1}{4}$	2	*Medium* (%)		*Hard* (%)	
							ROxf	RPar	ROxf	RPar
✔	-	-	-	-	-	-	83.3	89.4	66.3	79.2
✔	✔	✔	-	-	-	-	85.5	91.8	70.6	83.3
✔	✔	✔	✔	✔	-	-	86.4	93.2	72.6	85.6
✔	✔	✔	✔	✔	✔	✔	86.7	93.2	73.3	85.9

Table 4.6: Retrieval results on ROxf and RPar when considering different scales.

4.6.6 Impact of re-ranking

The impact of re-ranking on the proposed co-attention enabled CBIR pipeline is explored in this subsection. Two different re-ranking methods: α-weighted query expansion (αQE) [98] and diffusion [51] are considered.

αQE acts on feature vectors of top-ranked n images from the initial retrieval result by applying weighting averaging and re-normalization. The weight of the i-th ranked image descriptor is defined by $(\mathbf{V}_q{}^{\mathsf{T}}\mathbf{V}_i)^{\alpha}$ where $\mathbf{V}_q$ and $\mathbf{V}_i$ are the global feature vectors corresponding to the query image and the i-th ranked image. The aggregated feature vector serves as a query descriptor for the second-round retrieval and produces the final retrieval result.

Diffusion [51] is another powerful re-ranking method and has been applied in CBIR works [110]. Diffusion could be treated as an extension of the query expansion. Instead of only utilizing top n images as query expansion, based on first round retrieval results, diffusion explores the nearest neighbors by building a connection graph with the similarity score between each pair of images from the whole database for re-ranking.

The retrieval results of the baseline GeM (GeM†) and the proposed co-attention method (GeM†+CA) with these re-ranking methods are presented in Table 4.7. The proposed method GeM†+CA always gives better retrieval accuracy with or without the re-ranking. Specifically, on ROxford *Hard* set, even with re-ranking, the GeM† is still outperformed by GeM†+CA without any re-ranking. As visualization examples shown in Figure 4.5, the reason why simple GeM, which implicitly learns a query non-sensitive L2 norm attention at the training stage, not working is not because the query information is not comprehensive

enough, but because simple the query non-sensitive feature extraction manner will look at the wrong place in the image for feature extraction. It happens especially when the target object is not salient or surrounded by distractors. This problem can only be solved by a proper query sensitive attention mechanism, which would force the model to look towards the regions that match the query content, namely the co-attention.

Method	*Medium (%)*				*Hard (%)*			
	ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
R101-GeM†	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
R101-GeM†+αQE	84.4	77.8	91.7	82.7	68.8	56.2	82.8	65.9
R101-GeM†+DF	85.6	79.9	91.9	84.3	69.4	60.1	85.3	69.3
R101-DELG†	82.4	73.0	90.1	78.0	65.2	50.1	80.6	59.2
R101-DELG + SP†	84.1	75.9	91.0	79.2	68.8	53.6	83.0	62.3
R101-GeM†+CA	86.4	79.3	93.2	81.8	72.6	59.9	85.6	64.1
R101-GeM†+CA+αQE	86.9	79.6	93.3	84.5	72.8	60.2	85.7	68.7
R101-GeM†+CA+DF	87.2	81.1	94.4	86.1	73.7	63.9	88.4	72.0

Table 4.7: Retrieval results on ROxf and RPar datasets with re-ranking. For comparision, the DELG[14] with spatial verification (SP) re-ranking is also presented.

4.6.7 Impact of query noise

The standard evaluation protocol of ROxf/RPar dataset provides the bounding box for each query image. By default, all existing works would utilize the bounding box to crop the query image and only use the resulting patch as query input. One major concern for the proposed co-attention method is: whether it is over-fitting to ROxf/RPar dataset evaluation protocol or is robust enough when the query image is not cropped, containing noises and clutters. Figure 4.9 visualizes the co-attention map when not cropping the query image. When comparing the co-attention with and without the query crop, it can be observed that there is not much difference in the results, even though the query image in the second row contains a lot of background noise. Following the discussion from Section 4.2.1, spatial pooling implicitly implements an L2 norm attention mechanism. Within the proposed co-attention method pipeline, the query image features are selected based on their L2 norms before global pooling. Thus it has strong robustness to background noises that are irrelevant to training data, such as humans, grass, sky, and street, from the query image.

We do not consider the situation that one query image contains more than one potential

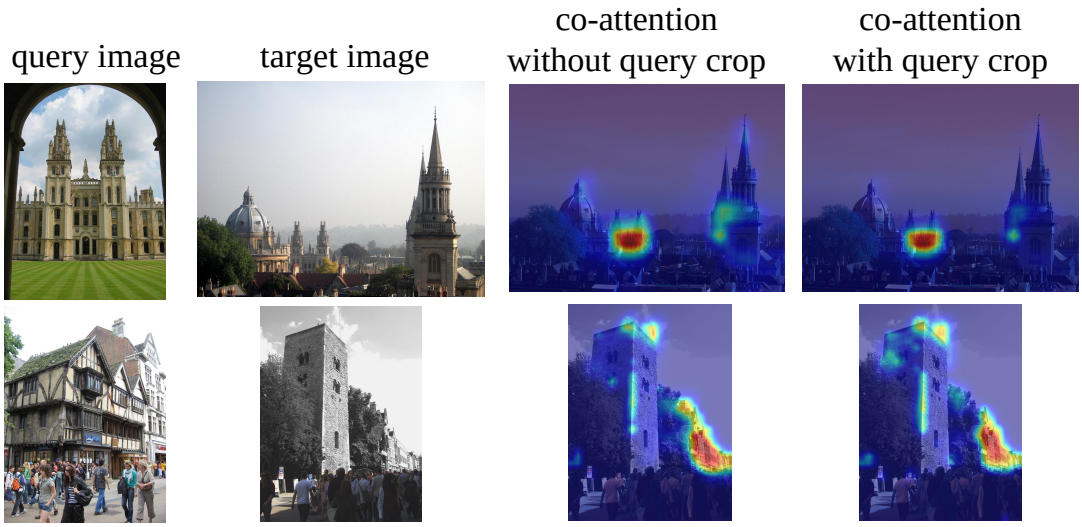


Figure 4.9: Co-attention visualization without query crop.

object of interest. On the one hand, the benchmark datasets ROxf/RPar [96] do not include this kind of situation. On the other hand, the search purpose is totally subjective to the CBIR system user. If the input query image contains multiple potential objects of interest, the user is supposed to specify which exact object (or region) needs to be searched, as the CBIR system can not know what the user has in mind. Suppose the user still would like to use a whole image that contains multiple training data relevant objects (regions) as query input and uniformly retrieve image content that matches with the query, as the example shown in Figure 4.10. In that case, the proposed co-attention will work similar to the query-nonsensitive attention, uniformly highlighting all training data relevant regions. This happens because the co-attention is guided by global feature of the query. If the query image contains multiple objects of interest, the resulting query global feature would be a mixture of feature representations of them, leading to an unfocused co-attention map in the end.

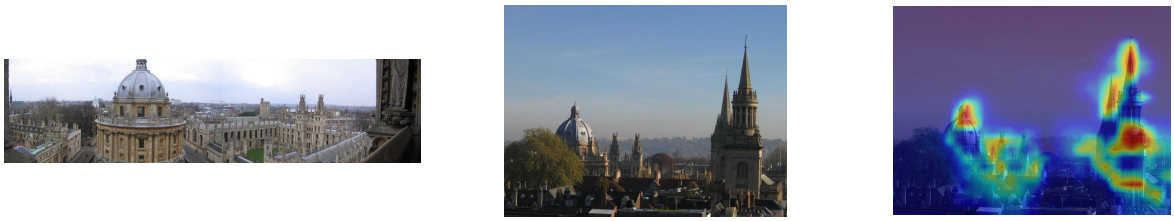


Figure 4.10: Co-attention visualization when consider a query image that contains multiple training data relevant object.

Table 4.8 provides the retrieval results of the baseline "GeM†", the proposed method "GeM†+CA" and the current state-of-the-art work DOLG with/without the query image crop. It can be seen that with or without query crop the co-attention method always improves the baseline GeM model's performance.

Method	query crop	*Medium (%)*				*Hard (%)*			
		ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
GeM†	✗	82.5	78.4	90.7	81.3	62.9	56.1	81.0	65.1
GeM†	✔	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
DOLG [146][8]	✗	83.2	79.0	91.6	82.9	64.8	57.9	82.6	67.3
DOLG [146][8]	✔	82.3	73.6	90.9	80.4	64.9	51.6	81.7	62.9
GeM†+CA	✗	85.5	81.8	93.6	83.9	69.2	61.4	85.8	67.7
GeM†+CA	✔	86.4	79.3	93.2	81.8	72.6	59.9	85.6	64.1

Table 4.8: Retrieval results on ROxf and RPar datasets without query crop.

4.6.8 Robustness to baseline model training

The previous good retrieval results are all based on using the GeM model pre-trained on the GLDv2 dataset with large batch size and ArcFace margin loss. What if the baseline model is trained with a much smaller dataset and simple loss? Will, in this case, the proposed co-attention method still provide positive improvements to retrieval performance? To test robustness to baseline model training, another GeM baseline model is trained following training settings from the original GeM pooling paper [98]. In details, the new GeM baseline model is trained on rSfM-120k dataset [98] which only contains around 90,000 images. The model is optimized with the simplest contrastive loss [24]. The batch size is set to 5. Each batch contains 5 image tuples. Each tuple contains 1 query image, 1 positive match image and 5 negative match images. The hard sample mining is also performed according to the description in [98] and the model is trained for 100 epochs. The proposed co-attention method is applied with the new GeM model and Table 4.9 shows the experiment results. It can be seen that even with smaller training data and simpler loss functions, the proposed co-attention still brings positive effects on retrieval performance. Of course, the improvement is not as impressive as that in Table 4.1, because when considering smaller training data and a simple loss function for the training, the feature tensor output by the backbone network is not as well-learned as when considering the large dataset GLDv2 and the ArcFace loss for training.

backbone	co-attention	*Medium* (%)		*Hard* (%)	
		ROxf	RPar	ROxf	RPar
Res50	✗	62.6	75.4	39.9	53.1
Res50	✓	69.3	79.2	42.9	57.5
Res101	✗	66.8	78.9	41.8	55.2
Res101	✓	70.1	80.3	45.1	59.5

Table 4.9: Retrieval results on ROxf and RPar datasets when trained on rSfM dataset with contrastive loss.

4.6.9 Why not directly perform similarity measure in one-to-many manner

In the proposed co-attention method, the similarity scores between the query image global feature $\mathbf{V}_q$ and candidate image clustered local features $\mathbf{X}_{c,K}$ are used as co-attention scores to re-weight $\mathbf{X}_{c,K}$ then perform GeM pooling to get the final candidate image global feature $\mathbf{V}_c$. One concern about the proposed co-attention method could be the necessity of co-attention weighted pooling. Why not just perform the similarity measure with $\mathbf{V}_q$ and $\mathbf{X}_{c,K}$ in a one-to-many manner. Three different methods are tested to calculate the final image pair matching score from K local match similarity scores between $\mathbf{V}_q$ and $\mathbf{X}_{c,K}$. As shown in Table 4.10, "Max" means using the maximum one among K local match similarity scores as the final image pair match score, while "Mean" means calculating the average value of them as the final result. "SoftMax" means applying the SoftMax function over K local match scores and then performing a weighted sum over them. All these methods lead to much worse results than the co-attention pipeline explained in Section 4.3.2.

Method	*Medium* (%)		*Hard* (%)	
	ROxf	RPar	ROxf	RPar
Max	81.4	90.3	64.8	81.6
Mean	77.4	88.1	58.7	77.2
SoftMax	79.6	89.0	62.5	79.3
GeM†+CA	86.4	93.2	72.6	85.6

Table 4.10: Retrieval results on ROxf and RPar with different one-to-many match ways.

4.6.10 More discussion about inverted file indexing

The local features selected by the L2 norm but without clustering are used for inverted file indexing implementation. One concern about this practice is that: why not apply inverted file indexing with the clustered local features $\mathbf{X}_{c,K}$. Intuitively speaking, within the co-attention enabled CBIR pipeline, the inverted file indexing is a general coarse-level filter that tries to filter out candidate images that are highly unlikely to match with the query. When applying the inverted file indexing, we do not want to accidentally filter out candidate images that are ground-truth matched with the query. Accordingly, we apply the inverted file indexing with a large codebook of size $K_{cdb} = 65536$ while also considering a large enough number of local features $N = 500$ from each image. So that any candidate image local features shares visual words with any query local features, this candidate image will be considered later for the co-attention generation and similarity measure evaluation. Features from the clustered local feature set $\mathbf{X}_{c,K}$ have few counts but relatively high-level semantic meaning, which is too discriminative between images. According to experimental results, applying the inverted file indexing with clustered local features $\mathbf{X}_{c,K}$ will filter out almost all database images and lead to worse retrieval performance.

In addition, the visual word codebook has an important role for both inverted file indexing and image representation building in the current state-of-the-art work HOW [123], as the ASMK [121] method is used by HOW to build local feature representations based on the usage of the visual word codebook. On the contrary, in the proposed co-attention CBIR pipeline, the inverted file indexing only serves as a general coarse-level filter to initially pick out candidate images for later comparison. The accuracy of the proposed co-attention method and image representation building do not rely on this mechanism. With the setting described in Section 4.5.1, according to experimental results, the inverted file indexing can speed up the retrieval by filtering out around 70% of easy negative images from the database. In other words, when considering the evaluation dataset ROxf/RPar with 1 million distractor images, we only need to perform the similarity measure with around 300,000 database images for each query image. Meanwhile, it does not perform any similarity measure or image feature match and almost makes no difference to the retrieval accuracy (mAP). As shown in Table 4.11, the proposed method "GeM†+CA" gives almost the same mAP results across ROxf/RPar datasets no matter with or without the Inverted File Indexing (IVF). The only difference is that using the inverted file indexing would

speed up the online retrieval procedure by filtering out those easily identifiable negative database images.

Moreover, all hyper-parameter setting, like the codebook size $K_{cdb} = 65536$, for the inverted file indexing module is based on HOW [123] and not specifically optimized, as it already results in a good retrieval speed.

Method	IVF	*Medium (%)*				*Hard (%)*			
		ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
GeM†+CA	✗	86.4	79.3	93.1	81.8	72.7	59.9	85.7	64.1
GeM†+CA	✔	86.4	79.3	93.2	81.8	72.6	59.9	85.6	64.1

Table 4.11: Retrieval results on ROxf and RPar datasets with/without inverted file indexing.

4.6.11 Computation cost

Considering $K = 10$ clusters, feature dimension $D^{'} = 512$, the memory cost to cache one candidate image is $10 \times 512 \times 4$ Bytes ≈ 0.02 MB and it takes around 21GB to cache the whole ROxf/RPar database with the 1 million distractor set.

The feature extraction takes in average 240ms to cache one candidate image's local features with 5 scales, including the time cost for the local feature clustering. It could be time-consuming, especially for a large database, but it can be performed offline, and it is only done once. With pre-cached features and the inverted file indexing, searching on ROxf/RPar with the 1 million distractor dataset for one query image takes on average 530ms with the help of acceleration by an NVIDIA Tesla GPU.

Considering a pair of the query image and candidate image of size 512×512 in original scale as input, ResNet101(R101) [45] as the backbone network for GeM feature extraction, Table 4.12 (a) and (b) compare the time cost of the GeM model at the online retrieval stage with or without the proposed clustering-based co-attention. $K = 10$ clustered local features of the candidate image have been pre-cached at the offline stage. All experiments were performed 100 times, and we report the average time cost of each component in Table 4.12.

As we can observe, at the online retrieval stage, when considering the proposed clustering-

For query feature extract				For cluster-based co-attention	Cosine
Backbone(R101)	Feat select	Pool and whiten	PCA	Co-attention generate and re-weighted pool	similarity
16	0.63	0.19	0.04	0.28	0.03

(a) Time cost of each component within the proposed clustering-based co-attention pipeline at the online evaluation stage.

For query feature extract		Cosine
Backbone(R101)	Pool and whiten	similarity
16	0.2	0.05

(b) Time cost of each component within the GeM [98] pipeline at the online evaluation stage.

Table 4.12: Time cost analysis of each component within the co-attention enabled pipeline (a) and the original GeM pipeline (b) at the online evaluation stage. A pair of one query image and one candidate image serves as input. The time cost of each component is reported in milliseconds (ms).

based co-attention, the extra time cost is mainly caused by two steps: the local feature selection over query image local features and the co-attention generation along with the re-weighted pooling (corresponds to the dashed rectangle in Fig 4.2). In total, for one pair of the query image and candidate image, the proposed co-attention method takes around extra 1ms to get the final image match score.

Detailed computation cost requirements and comparison with other models, when processing the ROxf/RPar datasets with 1 million distractor images in a batch-wise manner, are provided in Table 4.13. The proposed method "GeM†+CA" requires a similar memory cost as DELG [14]. When it comes to the retrieval time cost, "GeM†+CA" takes longer than others when considering a Tesla GPU, especially slower than GeM and DOLG, because they are simple global feature methods in which each image is only represented by a single global feature vector and the similarity measure is as simple as just calculating the cosine similarity with the global feature vector. However, the proposed co-attention method provides the best retrieval performance.

Method	Device	Memory (GB) ROxf/RPar+1M	Retrieval time (ms) in average
HOW [123]	CPU	14	750
GeM [98]	Tesla GPU	8	250
DOLG [146]	Tesla GPU	2	220
DELG+SP [14]	Tesla GPU	22	383
GeM†+CA (ours)	Tesla GPU	21	530

Table 4.13: Computation cost comparison.

Compared with the CANet from Chapter 3, the major advantage of the clustering-based co-attention method is that it only extracts $K = 10$ clustered local features from each candidate image and the co-attention map is generated by simply performing cosine similarity measure between the query global feature and clustered candidate local features. This manner makes it feasible to pre-cache the clustered local features of the candidate images at the offline stage, with no need to feed candidate images through the deep neural network at the online stage.

4.7 Conclusion

In this chapter, we enable large-scale content-based image retrieval with co-attention mechanisms for the first time. The proposed co-attention method can be treated as a non-trainable-parameter module for a pre-trained spatial pooling model. It is intuitively based on the similarity score between the global feature vector of the query image and the clustered local features from the candidate image. The extra computation cost caused by the query sensitivity is addressed by employing local feature clustering while also considering the inverted file indexing to speed up the retrieval procedure. While straightforward, the proposed co-attention method generates good co-attention maps even in some challenging cases. By simply adding our co-attention method to the pre-trained baseline GeM model, the retrieval performance is greatly improved and results in a new state-of-the-art retrieval performance on benchmark datasets with comparable computation costs to existing models.

According to experimental results obtained by the proposed co-attention method, a conclusion can be drawn that performing clustering over local features from the convolution feature tensor could generate meaningful clustered local features, in which local features belonging to the same object regions will be automatically grouped together, as demonstrated in Figure 4.4. This approach not only dramatically reduces the number of local features extracted from each candidate image but also makes the resulting local features aware of more neighbour location information, representing an area over the target object instead of a simple grid local patch over the original image. These clustered local features are more expressive than each original entry on the convolution feature tensor but more localized than the naive global pooling feature. Then, co-attention maps are

derived from these well-extracted candidate images' local features by simply performing the cosine similarity measure with the query global feature.

Despite its success, the proposed clustering-based co-attention method still has room for improvement. One central problem is that the co-attention calculation must be performed at the online stage to build the weighted global feature for each candidate image before performing the similarity measure. This means for each candidate image, its feature must be cached as a real-value float number and it requires GPU acceleration at the retrieval stage. Although the usage of feature selection and clustering has made the extra computation cost comparable to existing works, there are some models that would further compress database image features with binary encoding, like HOW[123], which leads to significantly fewer memory requirements and would also eliminate the reliance of using a GPU device at the retrieval stage. In the research presented in the following chapter, we aim to maintain excellent performance improvement and good interpretable query-sensitive spatial attention while trying to further reduce the computation cost.

Chapter 5

Expressive local feature match

5.1 Introduction

Former chapters have proposed different methods for co-attention generation. Especially, the clustering-based co-attention could extract few but expressive clustered local features from each candidate image and generate good co-attention maps even under challenging situations. However, the co-attention generation procedure needs to be performed at the online retrieval stage. It means each element of the candidate image local features must be cached as a real-value float number for calculating the co-attention map. This fact not only causes extra computation cost at the online retrieval stage but also prevent the feature from being further compressed by binary encoding, which has been widely applied in other CBIR methods for cost reduction.

Based on the clustering-based feature extraction pipeline from the last chapter, this chapter explores further using the clustered local features to perform many-to-many local feature matching for content-based image retrieval. Unlike existing local feature methods, such as HOW[123] that tend to store huge amounts of low-dimensional local features and apply complex match kernel, like ASMK [121], for similarity measure, this work proposes a corresponding many-to-many similarity criterion apply on the few but expressive clustered local features. Moreover, as binary encoding has been widely applied for large-scale image search, we also propose a trainable binary encoding layer that is initialized with Principal Component Analysis (PCA) and then fine-tuned based on the idea of the Bi-half

Net [73]. After fine-tuning, the proposed binary encoding layer generates compact binary codes with slight performance degradation. According to the experimental results, the proposed local match method could achieve comparable CBIR accuracy to the clustering-based co-attention method from the last chapter but with much lower computation costs and does not require GPU acceleration at the online retrieval stage. Besides, extensive visualization results also demonstrate that the local match method implicitly leads to a co-attention-like local match map, in which the effectiveness and contribution of the local features for each candidate image varies with the query feature.

The rest of this chapter is organized as follows. Some preliminaries are introduced in Section 5.2. The proposed methodology of performing many-to-many local feature matching with expressive binary code for CBIR is explained in Section 5.3. The experimental results are provided in Section 5.4, while additional ablation studies and discussion are provided in Section 5.5. The conclusions of this chapter are drawn in Section 5.6.

5.2 Preliminary

In this section, we start with introducing the baseline GeM model and the Bi-half Net. Afterward, we present our proposed expressive local feature extraction pipeline and many-to-many local feature matching method for content-based image retrieval.

5.2.1 Baseline model structure and training

Similar to the model described in Section 4.2.2, in this chapter, we still use ResNet [45] as the backbone network followed by a generalized mean pooling layer defined by Eq. (4.1), with a fixed power co-efficient $p = 3$, and a trainable fully connected layer for feature whitening. The model is also trained with the ArcFace loss (Eq.4.4).

5.2.2 Binarization and Bi-half Net

The Sign function is a straightforward binary encoding module to transform a continuous real value space feature into a binary code, defined as:

$$sign(x) = \begin{cases} -1, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases} \tag{5.1}$$

The Sign function has been the default choice for some CBIR works to binarize continuous real-value features [123, 87, 14]. Nevertheless, its direct application over real-value features, which are optimized with the real-value loss, could lead to information loss, corrupting the whole model's performance. This happens because the continuous real-value feature representation reflects data-defining information by means of the variance across each dimension (channel). Suppose we simply replace a continuous real-value with a binary symbol (+1 or -1). In that case, some channels that originally had significant value differences might be arbitrarily quantized to the same code, losing the information they used to carry. This could be a severe problem for CBIR, as the most commonly used loss function, such as triplet loss or contrastive loss, and similarity measure metrics, such as L2 distance, are all based on real value variance across each channel calculation. For example, let us consider two dimensions (channel) toy features: $[0.8, 0.1]$ and $[0.2, 0.9]$; they have a significant L2 difference so that the model can distinguish between them. However, after the binarization with the Sign function, they are both represented by $[1, 1]$. Thus, the binary code losses all discriminative information for these two features.

Accordingly, a good binary encoder is supposed to transmit as much information as possible after binarizing a continuous feature representation. What exact attributes make a good binary code has been discussed in several works [141, 135]. Especially, the recent work Bi-half Net [73] explained this from the aspect of information theory. According to the derivation in [73], the information per channel transmitted from the original continuous features to the corresponding binary code is maximized when the binary value (-1, 1) distribution across all channels is half-half between the codes of 1 and -1 :

$$p(B = 1) = P(B = -1) = \frac{1}{2}. \tag{5.2}$$

To achieve this goal, at the training stage, before feeding a batch of feature vectors $\mathbf{F}$ into any loss function $\mathcal{L}$, a Bi-half layer π_0 is applied to transform $\mathbf{F}$ into the binary code $\mathbf{B}$ while guaranteeing that each channel has equal probability to be -1 or $+1$:

$$\mathbf{B} = \pi_0(\mathbf{F}) = \begin{cases} 1, & \text{top half of sorted } \mathbf{F} \\ -1, & otherwise \end{cases} \tag{5.3}$$

A major problem of directly optimizing the binary codes is the vanishing gradient as the binarization operation is not differentiable. To obtain the gradients for the back-propagation training, a straight through strategy [10, 116] is applied for the hash layer's gradient calculation. Additionally, the continuous feature $\mathbf{F}$ distribution is supposed to align with the ideal half-half distributed binary code $\mathbf{B}$ [73]. Thus the final forward and backward process of the Bi-half layer [73] is defined by:

$$\begin{aligned} &\text{Forward: } \mathbf{B} = \pi_0(\mathbf{F}), \\ &\text{Backward: } \frac{\partial \mathcal{L}}{\partial \mathbf{F}} = \frac{\partial \mathcal{L}}{\partial \mathbf{B}} + \varphi(\mathbf{F} - \mathbf{B}), \end{aligned} \tag{5.4}$$

where φ is a hyper-parameter and it equals the multiplicative inverse of the element count of feature batch $\mathbf{F}$.

5.3 Extracting expressive binary local features

In the following, we describe how we extract compact but expressive binary local features from the feature tensor output by the backbone network and perform many-to-many local feature matching for the CBIR task.

5.3.1 Local feature extraction

The first challenge when performing many-to-many local feature matching is still the computation cost caused by large numbers of non-informative local features extracted from a given image. As explained in Section 4.6, given an input image $\mathbf{I}$ of size $h \times w$, after feeding

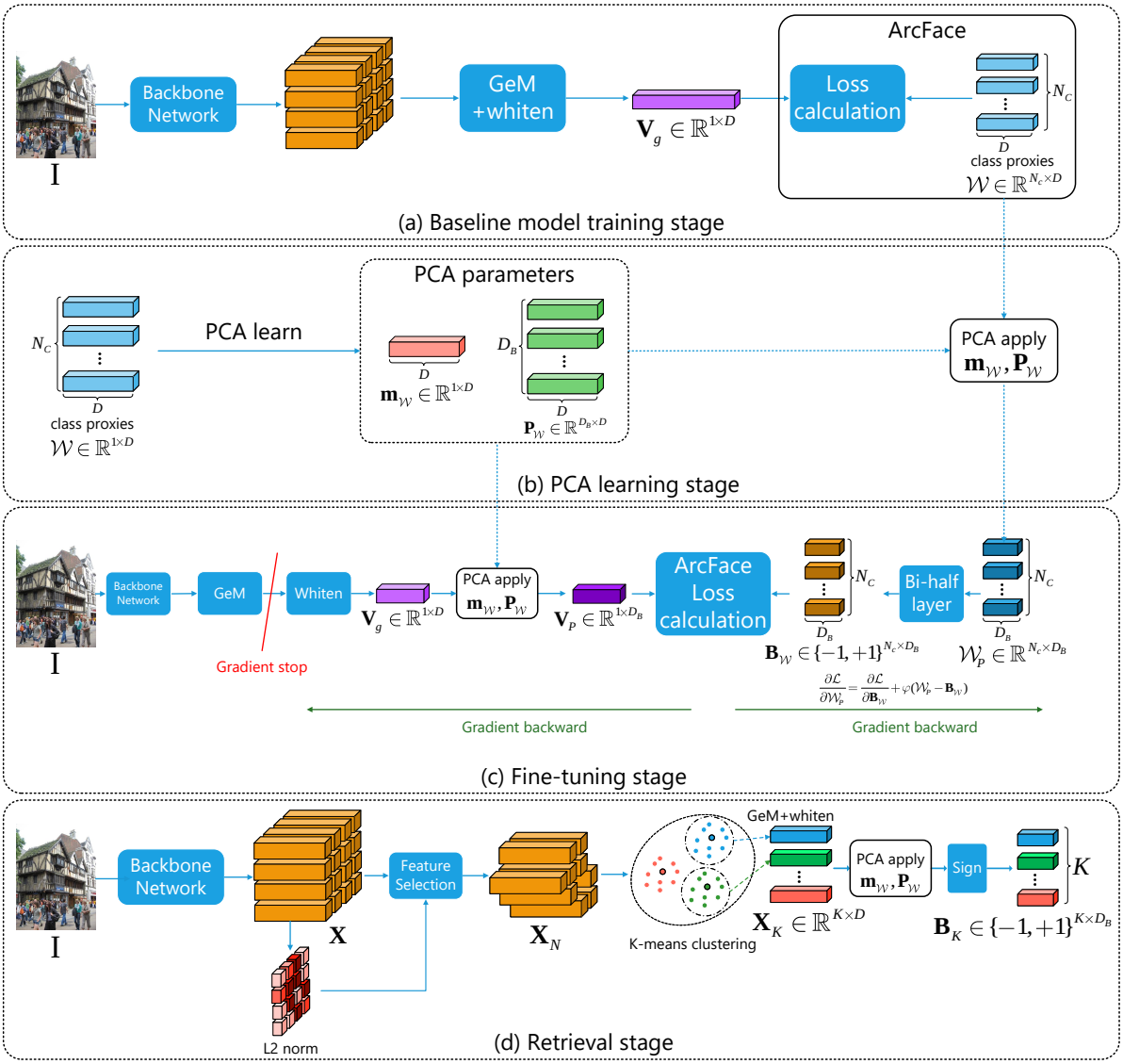


Figure 5.1: Illustration of the local match method at different stages. (a) training the baseline GeM model follows the description from Section 4.2.2. (b) learn the PCA projection parameters with pre-trained ArcFace class proxy features. (c) fine-tuning the PCA projection parameter and dimension reduced class proxies with bi-half layer. (d) expressive local feature representation building at retrieval stage.

through ResNet [45], hundreds of local features could be extracted and the number could exponentially increase if consider multi-scale feature extraction scheme. Fortunately, the success of co-attention, described in Chapter 4, has proved that combining L2 norm based feature selection and simple k-means clustering could result in few but meaningful clustered local features for the input image. Accordingly, with the input image $\mathbf{I}$ as input, after feeding through the backbone network, L2 norm based feature selection followed by k-means clustering are performed, resulting in selected local features $\mathbf{X}_N \in \mathbb{R}^{N \times D}$ and clustered local features $\mathbf{X}_K \in \mathbb{R}^{K \times D}$ $(K << N)$ separately.

5.3.2 Local feature compression and fine-tuning

In the following, we discuss dimension reduction and binarization for further feature compression and computation cost reduction.

Dimension reduction by PCA. The dimension reduction is still applied with PCA. Normally, PCA serves as post-processing after the CNN has been trained and the PCA parameters (the projection matrix for dimension reduction) are learned with features of random sample images from the training dataset. As mentioned in Section 4.2.2, the ArcFace weight matrix $\mathcal{W} \in \mathbb{R}^{N_c \times D}$ from Eq. (4.4) could be treated as proxy features for each image class. We use $\mathcal{W}$ as a set of comprehensive sample features which are more informative than those extracted from a set of random training images for the PCA parameter learning. The resulting learned PCA parameters: mean and eigenvectors, are denoted as $\mathbf{m}_{\mathcal{W}} \in \mathbb{R}^{1 \times D}$ and $\mathbf{P}_{\mathcal{W}} \in \mathbb{R}^{D_B \times D}$, where D corresponds to the original backbone network output dimension, D_B is a hyper-parameter, indicating the reduced output feature dimension. For any new incoming features $\mathbf{Y} \in \mathbb{R}^{D}$, the dimension reduced output feature $\mathbf{Y}' \in \mathbb{R}^{D_B}$ is calculated as:

$$\mathbf{Y}' = (\mathbf{Y} - \mathbf{m}_{\mathcal{W}})\, \mathbf{P}_{\mathcal{W}}^{\mathsf{T}} \tag{5.5}$$

Binarization with Bi-half fine-tuning. Binary encoding is another common practice to further compress the extracted features. However, as mentioned before, by directly applying the Sign function for feature binarization would result in information loss. Experimental results (see the results from Section 5.5) indicate a negative impact on the results when simply applying the Sign function together with the PCA feature dimension reduction. To address this problem, as the PCA operation defined by Eq. (5.5) is differentiable, we adapt the Bi-half layer from the Bi-half Net [73] to fine-tune both the fully-connected whitening layer and the PCA parameters: $\mathbf{m}_{\mathcal{W}}$ and $\mathbf{P}_{\mathcal{W}}$. As shown in Figure. 5.1 (c), during the fine-tuning stage, we have a dimension reduction of the class proxy features from the ArcFace loss module by the learned PCA parameters $\mathbf{m}_{\mathcal{W}}$, $\mathbf{P}_{\mathcal{W}}$. The resulting features $\mathcal{W}_P$ serve as the new class proxy features at the fine-tuning stage. For each input image $\mathbf{I}$ at the fine-tuning stage, after whitening by the fully connected layer, PCA dimension reduction with parameters $\mathbf{m}_{\mathcal{W}}$, $\mathbf{P}_{\mathcal{W}}$ are also applied on the global

GeM feature $\mathbf{V}_g$ resulting in a compact feature vector $\mathbf{V}_P \in \mathbb{R}^{1\times D_B}$.

The original Bi-half layer is applied in a batch-wise manner. In other words, it actually enforces that each channel within each batch has equal probabilities to be -1 or $+1$. However, what we actually want is to have a half-half distribution of -1 and $+1$ bits over the whole binary feature space. At the training stage, it is impossible to have the batch size as large as the whole training database; the batch size setting may also vary with the training procedure implementation and GPU hardware memory capacity, making the training procedure more unstable. For example, let us consider a set of toy features with batch sizes of 2: $[0.5, 0.3]$, $[0.1, 0.9]$. Applying the Bi-half layer on this batch of features will result in the binary codes: $[1, -1]$, $[-1, 1]$. However, if the batch features would change to: $[0.5, 0.3]$, $[0.6, 0.1]$, then the resulting Bi-half code will be $[-1, 1]$, $[1, -1]$. Only due to changes to other features within the same batch lot, the binary code for the same feature $[0.5, 0.3]$ could change from $[1, -1]$ to $[-1, 1]$. Nevertheless, the ArcFace loss works with the class proxy features $\mathcal{W}$. For a large enough training dataset (like GLDv2 [136]), the number of classes is much larger than in the common batch size setting. Each class proxy feature represents a whole class of images instead of a single image. Enforcing these proxy features to have a half-half binary symbol distributions could potentially make the binary code of the same class images be optimized towards a consistent goal across all batch steps and eliminate the distraction caused by the batch size setting or the random image sample shuffle at the training stage. Accordingly, as shown in the middle part of Figure. 5.1 (c), we apply the Bi-half layer on the dimension reduced proxies $\mathcal{W}_P$ to get the binarized proxies $\mathbf{B}_{\mathcal{W}}$. $\mathbf{V}_P$ and $\mathbf{B}_{\mathcal{W}}$ are used to calculate the ArcFace loss and then Eq. (4.4) is re-written as :

$$L(\widehat{\mathbf{V}}_P, \mathbf{y}) = -\log\left(\frac{\exp(\gamma \times \mathrm{AF}(\widehat{\mathbf{V}}_P\widehat{\mathbf{b}}_i^{\mathsf{T}}, y_i))}{\sum_{j=1}^{N_c}\exp(\gamma \times \mathrm{AF}(\widehat{\mathbf{V}}_P\widehat{\mathbf{b}}_j^{\mathsf{T}}, y_j))}\right), \tag{5.6}$$

where $\widehat{\mathbf{V}}_P$ is the L2 normalized image feature vector $\mathbf{V}_P$. $\widehat{\mathbf{b}}_i$ refers to the proxy feature for class i from the ArcFace weight matrix $\mathbf{B}_{\mathcal{W}}$. Gradients of the Bi-half layer are obtained according to Eq. (5.4).

It needs to point out that, at the training stage, the Bi-half layer would only re-assign the value of each feature element to either -1 or $+1$ as floats and not as binary bits, because we have to perform backward transmission based on Eq. (5.4). In other words, the fine-

tuning procedure is to encourage $\mathcal{W}_P$ to approach the half-half distributed binary values $\{-1, 1\}$ while forcing each training image feature $\mathbf{V}_P$ to get close to the corresponding binary class proxy feature from $\mathbf{B}_\mathcal{W}$ and move away from others.

In addition, starting from the ArcFace loss, the gradient is only transmitted backward to the proxies $\mathcal{W}_P$ and right before the backbone network, as shown in Figure. 5.1 (c). In other words, we only fine-tune the fully connected whitening layer, PCA parameters ($\mathbf{m}_\mathcal{W}$ and $\mathbf{P}_\mathcal{W}$) and the new class proxies $\mathcal{W}_P$. There are two reasons for this: 1) as the backbone network has been well-trained at the baseline model training stage, we would like to train a binary coding module to transform existing well-learned continuous features into informative binary codes. Freezing the backbone network could help us ensure a fair comparison during the ablation study experiments and make sure the performance improvement is only caused by the implementation of binary encoding and local feature match, instead of some training tricks. 2) freezing the backbone network and only fine-tuning the other trainable parameters would ensure significant speed up of training procedure. More detailed discussion is provided in Section 5.5.3.

In a way, the fully connected whitening layer and PCA parameters $\mathbf{m}_\mathcal{W}$, $\mathbf{P}_\mathcal{W}$ together work as a trainable encoder to project the original GeM pooled features into a D_B dimension latent space. This enforces that each feature channel has an equal probability to be 1 (larger than 0) or -1 (smaller than 0). As a result, at the retrieval (evaluation) stage, after each real-value feature element is binarized by the Sign function, we can keep as much information (variance) of each channel as possible, leading to a good performance.

5.3.3 Local feature match

Let us consider a pair of images, representing the query image $\mathbf{I}_q$ and the candidate image $\mathbf{I}_c$ from a given database. After the feature extraction pipeline as shown in Figure. 5.1 (d), two corresponding binary coded local feature sets $\mathbf{B}_{q,K} = \{\mathbf{b}_{q,i} | i = 1, \ldots, K\}$ and $\mathbf{B}_{c,K} = \{\mathbf{b}_{c,j} | j = 1, \ldots, K\}$ are extracted. Then, a similarity matrix $\mathbf{M} = [m_{i,j}] \in \mathbb{R}^{K \times K}$ is obtained by calculating the similarity score between each pair of query local feature $\mathbf{f}_{q,i}$ and local feature candidates $\mathbf{f}_{c,j}$:

$$m_{i,j} = d(\mathbf{b}_{q,i}, \mathbf{b}_{c,j}), \tag{5.7}$$

where $d(\cdot,\cdot)$ is a similarity function. For real number feature vectors, it could be the cosine similarity. For binary features, a common way of measuring the distance between two binary coded sequences is by calculating the Hamming distance. Hamming distance represents the count of those bits that differ between two vectors. The Hamming distance ranges between $[0, D]$ given that a vector has a feature dimension D and it can be normalized to the range $[0, 1]$ by dividing with the feature dimension D. We evaluate the similarity by subtracting the Hamming distance from 1, given that the Hamming distance indicates the difference, which is the counterpart of similarity. Consequently, the i-th row of the similarity matrix $\mathbf{M}$ stores the similarity score between $\mathbf{b}_{q,i}$ and each local feature from the candidate image feature set $\mathbf{B}_{c,K}$.

In principle, the matrix $\mathbf{M}$ is supposed to be transformed into a single similarity score between the image pair $\{\mathbf{I}_q, \mathbf{I}_c\}$ and the similarity score calculation is supposed to be computed fast online. Thus, we first define the similarity score between a single query local feature $\mathbf{b}_{q,i}$ and the whole candidate image by:

$$s(\mathbf{b}_{q,i}, \mathbf{I}_c) = \max_j m_{i,j}, \tag{5.8}$$

and eventually, the similarity between images $\mathbf{I}_q$ and $\mathbf{I}_c$ is given by :

$$S(\mathbf{I}_q, \mathbf{I}_c) = \frac{\sum_{i=1}^{K} s(\mathbf{b}_{q,i}, \mathbf{I}_c)}{K}. \tag{5.9}$$

As explained in Section 4.4.2, here we also apply the inverted file indexing for online retrieval speed up. The framework of the proposed method when considering the inverted file indexing is shown in Figure 5.2. Most details are the same to descriptions in Section 4.4.2. The main differences are: at the feature caching stage, with the selected local features $\mathbf{X}_{c,N}$, after performing k-means clustering and PCA dimension reduction with fine-tuned components $\mathbf{m}_{\mathcal{W}}$ and $\mathbf{P}_{\mathcal{W}}$, it is binarized by Sign function, resulting in the clustered binary local features $\mathbf{B}_{c,K}$. At the online retrieval stage, the clustered local features $\mathbf{X}_{q,K}$ is also PCA dimension reduced and binarized by the Sign function. Then, according to the cached dictionary, only those database images that share at least one visual word with the query image are picked out for the later local feature matching and similarity measure as described in Section 5.3.3.

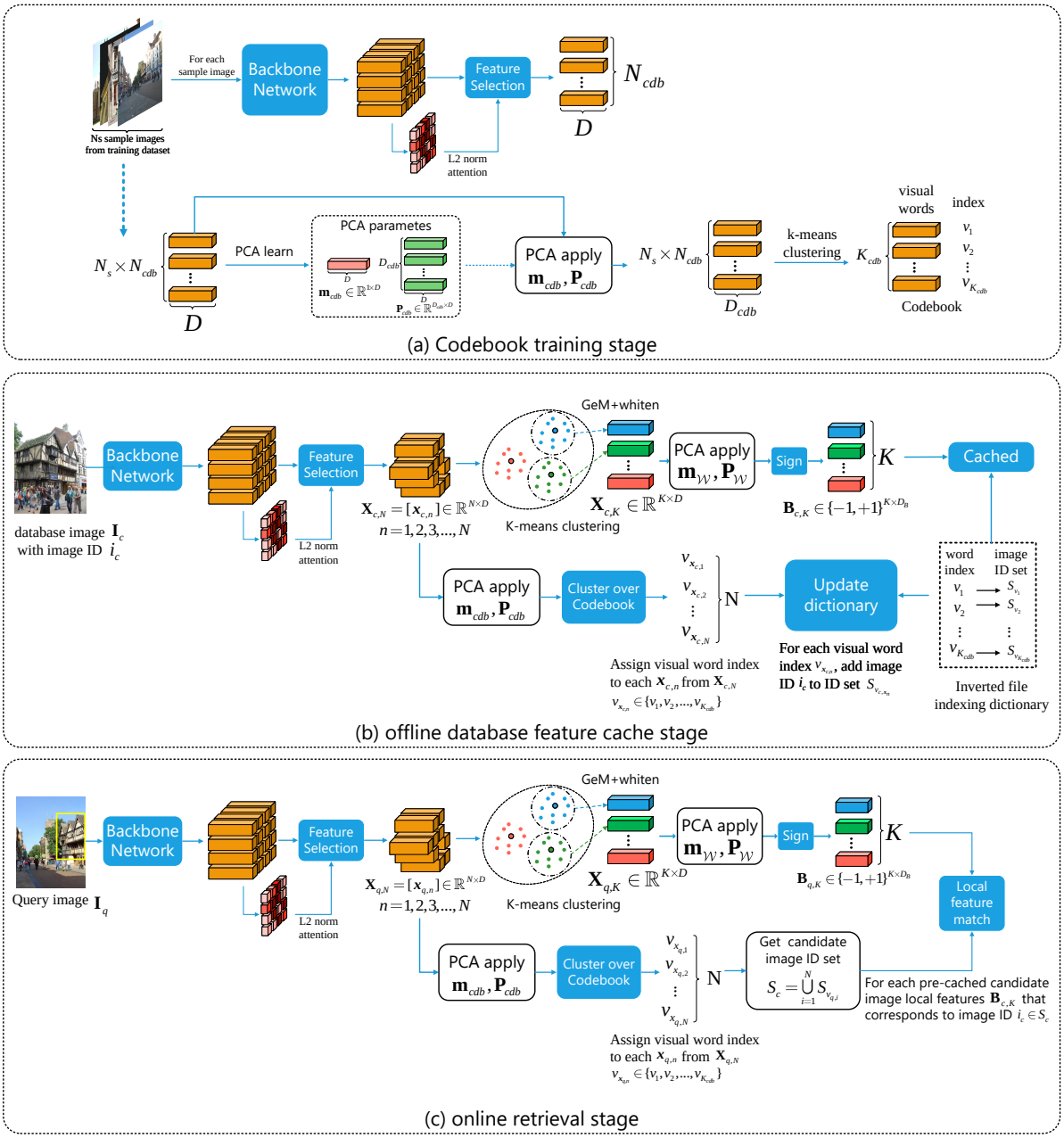


Figure 5.2: Illustration of the proposed method's pipeline with inverted file indexing.

5.4 Experiments

In this section, we first discuss the experiment setup. Then, some visualization results are presented to show the effeteness of the proposed local feature match strategy. After that, retrieval results comparison between the proposed local match method and other existing state-of-the-art works are provided.

5.4.1 Experimental setup

Implementation details. We still consider ResNet101 (and ResNet50) [45] as the backbone network ($D = 2048$). Settings for the baseline GeM model training and inverted file indexing are the same as description in Section 4.5.1 For the Bi-half fine-tuning, we use the same training setting as the baseline GeM model, except that the initial learning rate is set to 0.0001. The GeM baseline model is trained for 50 epochs and then fine-tuned with Bi-half layer for 10 epochs.

At the retrieval stage, we still set the local feature selection count $N = 500$ and the cluster number $K = 10$ for k-means clustering. After whitening, the clustered local features of both query image and candidate image, represented by $\mathbf{X}_{q,K}$ and $\mathbf{X}_{c,K}$, are compressed using the PCA dimension reduction with fine-tuned parameters $\mathbf{m}_{\mathcal{W}}$ and $\mathbf{P}_{\mathcal{W}}$ to $D_B = 512$, followed by binarization with the Sign function, given by Eq. (5.1).

We also consider the multi-scale scheme feature extraction, as explained in Section 4.4.3, when not considering local feature match we use 3 scales $\{1, \sqrt{2}, \frac{1}{\sqrt{2}}\}$. When considering the proposed local feature match method, we still have feature selection from 5 scales: $\{\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$.

Evaluation dataset. Revisited Oxford and Paris datasets [96] along with the 1 million disctrator set R1M [96] are still considered as evaluation datasets in this chapter. By default, each input query image is cropped with the provided bounding boxes for standard evaluation protocol.

5.4.2 Local match visualization

As mentioned in Section 5.3.3, we only keep the most similar candidate image clustered local feature as its match for each clustered query local feature. Match scores for the locations that are not selected are set to zero. Local features grouped into the same cluster share the corresponding clustered local feature as their representation. Match scores of all local features from images at different scales are projected back to the corresponding location (or region) from the original image and accumulated to get the final score map.

We consider heatmaps to show the effectiveness of the proposed local match method. Given a pair of query and candidate images as shown in Figure. 5.3 (a), while each row from Figure. 5.3 (b) shows two pairs of matching clustered local features from the query image and candidate image. Each pair of matching clustered local features results in a localized heatmap indicating the relevant regions. We have 10 clusters for each image. Thus we obtain 10 local matching maps as shown in Figure. 5.3 (b), exampled (1) to (10). Location-wise adding all these local match heatmaps and normalizing to range $[0, 1]$ would finally get the global match map as shown in Figure. 5.3 (c).

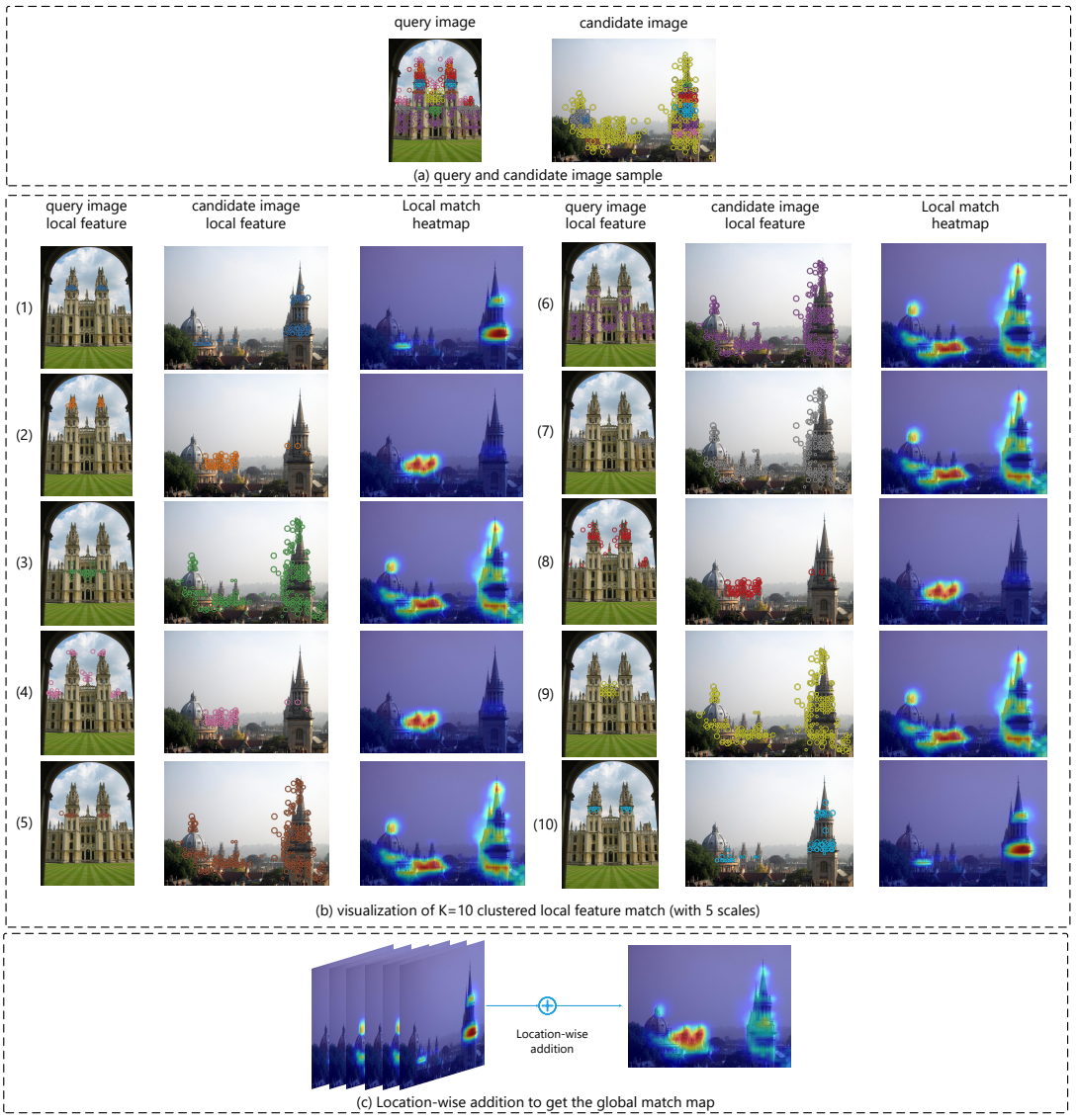


Figure 5.3: Visualization of the local matching examples.

In Figure. 5.4, we show more examples of local match map visualization results along with their corresponding L2 norm attention maps for comparison. The L2 norm reflects the importance of each location, as discussed in Section 4.2.1, or how much it contributes to the final feature vector obtained by global pooling. Accordingly, the L2 norm attention

map also reflects each location's contribution to the similarity in the query-target image pair. As we can observe, the L2 norm attention maps tend to be evenly distributed over the relevant content of the training data (in this case represented by all landmarks or buildings content). For some easy cases, when the target region is salient and of large scale, it may work well. However, for some hard cases with multiple training data relevant objects, like the five examples from Figure. 5.3, the L2 norm would not be able to choose the correct location to focus on and could highlight many unwanted regions or even indicate the wrong places. On the contrary, with the proposed local match implementation, most matching local feature pairs between the query and target images would have the highest similarity score. As a result, these matching local feature locations would also represent the most important contributions to the final similarity score between the image pair. When considering the same target image with different query content, as in examples 5 and 6, the result local match maps correctly highlight the corresponding regions of interest. In a way, the visualization of local match maps looks like co-attention, as the importance of each local feature from the candidate image is no longer fixed as in the traditional global spatial pooling. The effectiveness and contribution of the local features for each candidate image varies with the query feature. In each candidate image from Figure. 5.4, the local match score comes mostly from the region that matches the query content, even when the target object is not salient or is surrounded by some other similar class objects.

5.4.3 Retrieval results

Quantitative result. Image retrieval results of existing works and our local match method are provided in Table 5.1. As mentioned in Section 4.5.4, for fair comparison, some recent state-of-the-art works are re-implemented and marked with "†".

Group (A) from Table 5.1 shows the results of local feature methods. Group (B) from Table 5.1 shows the result of the global feature methods while the bottom group (C) from Table 5.1 shows the results of the proposed local match method with PCA dimension reduction and Bi-half fine-tuning applied (LM-BiHalf). The local match method could be treated like a post-processing module over the pre-trained baseline GeM model. As we can observe, it significantly improves the baseline model's retrieval performance. Especially,

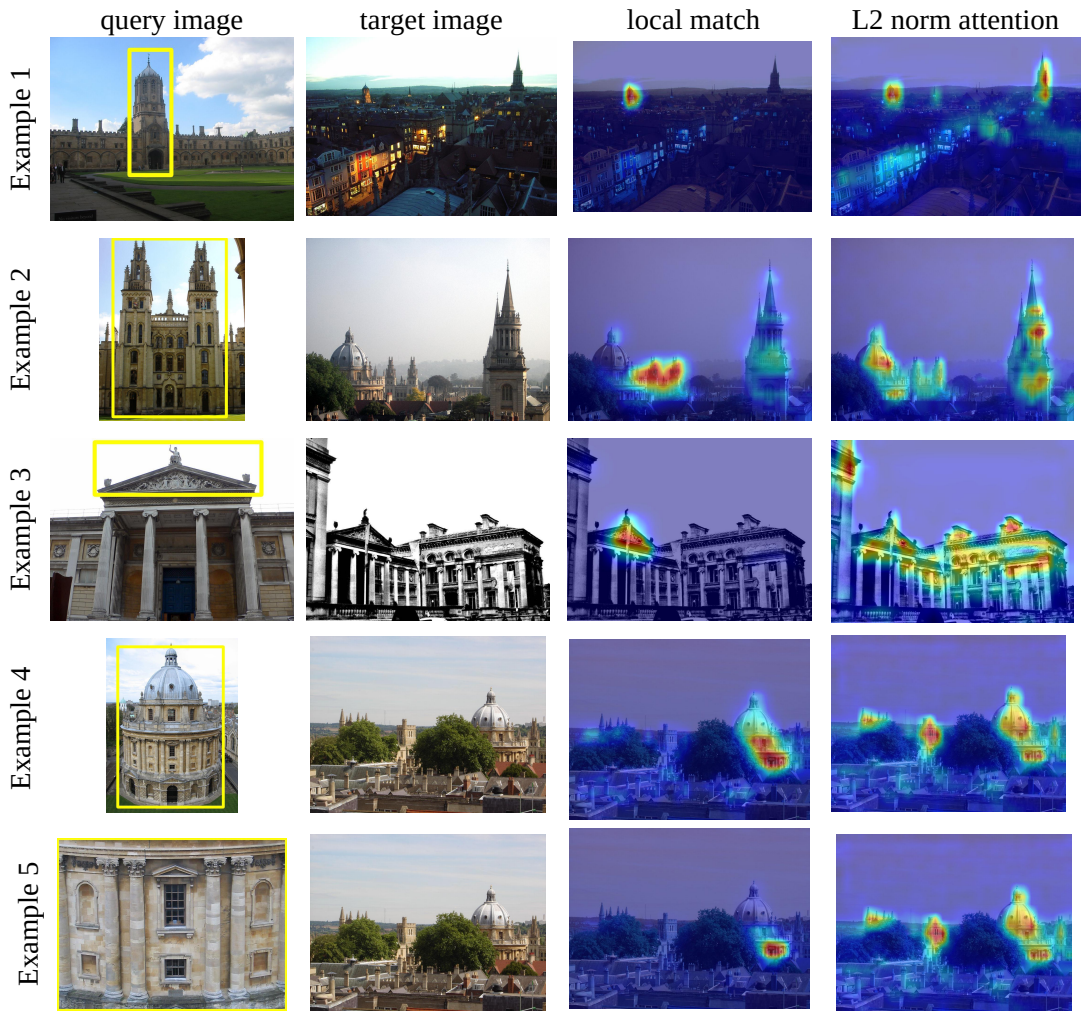


Figure 5.4: Visualization of the proposed local match and comparison with L2 norm attention.

when considering the ResNet101 as the backbone network, on the *Hard* set of ROxf (RPar), the mAP of local match method reaches 72.0% (83.6%). Moreover, when considering the 1 million distractor set, the local match method still outperforms current state-of-the-art works DELG and DOLG on ROxf+1M dataset and show comparable results on RPar+1M dataset.

5.4.4 Qualitative retrieval results

In Figure. 5.5, we provide a qualitative retrieval results comparison between the proposed local match method "GeM†-LM-BiHalf" and the baseline GeM model on the challenging ROxf dataset [96], considering the *Hard* evaluation protocol. The query image is shown on the first column from the left side of each row with a yellow bounding box indicating

Method	*Medium (%)*				*Hard (%)*			
	ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
(A) Local feature								
HesAff-rSIFT-ASMK*+SP [121]	60.6	46.8	61.4	42.3	36.7	26.9	35.0	16.8
HardNet-ASMK*+SP [79]	65.6	-	65.2	-	41.1	-	38.5	-
DELF-ASMK*+SP [120]	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
DELF-D2R-R-ASMK*+SP [120]	76.0	64.0	80.2	59.7	52.4	38.1	58.6	29.4
R50^{-}-HOW-MDA [137]	82.0	68.7	83.3	64.7	62.2	45.3	66.2	38.9
R50^{-}-HOW [123]	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
R101^{-}-HOW (GLDv2)†	83.9	77.9	87.9	76.4	71.3	52.8	76.0	56.4
(B) Global feature								
R101-R-MAC [39]	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
AlexNet-GeM [98]	43.3	24.2	58.0	29.9	17.1	9.4	29.7	8.4
VGG16-GeM [98]	61.9	42.6	69.3	45.4	33.7	19.0	44.3	19.1
R101-GeM [98]	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
R101-GeM-AP [101]	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
R101-GeM↑ [110]	67.3	49.5	80.6	57.3	44.3	25.7	61.5	29.8
R101-GeM (GLD) [85]	67.3	49.5	80.6	57.3	44.3	25.7	61.5	29.8
R101-DSM [110]	65.3	47.6	77.4	52.8	39.2	23.2	56.2	25.0
R101-SOLAR [85]	69.9	53.5	81.6	59.2	47.9	29.9	64.5	33.4
R50-DELG [14]	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
R50-DELG + SP [14]	78.3	67.2	85.7	69.6	57.9	43.6	71.0	45.7
R101-DELG [14]	76.3	63.7	86.6	70.6	55.6	37.5	72.4	46.9
R101-DELG + SP [14]	81.2	69.1	87.2	71.5	64.0	47.5	72.8	48.7
R101-DELG†	82.4	73.0	90.1	78.0	65.2	50.1	80.6	59.2
R101-DELG + SP†	84.1	75.9	91.0	79.2	68.8	53.6	83.0	62.3
R50-DOLG [146][8]	81.2	71.4	90.1	79.0	62.6	47.3	79.2	59.8
R101-DOLG [146][8]	82.3	73.6	90.9	**80.4**	64.9	51.6	81.7	**62.9**
(C) Our method								
R50-GeM†	79.8	69.0	87.3	73.1	60.4	44.2	74.0	52.0
R50-GeM†-LM-BiHalf	84.4	72.4	91.0	74.8	67.9	50.7	81.6	53.9
R101-GeM†	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
R101-GeM†-LM-BiHalf	**86.7**	**76.6**	**92.0**	79.3	**72.0**	**54.8**	**83.6**	61.4

Table 5.1: Image retrieval results on ROxf/RPar datasets (and their extended version +1M distractor set R1M), considering *Medium* and *Hard* evaluation protocols. Groups (A) and (B) separately show the results of local and global feature methods. The bottom group (C) shows the results of our model. "GeM†" means the re-implemented baseline GeM model, which is trained with the setting from Section 5.4.1. "GeM†-LM-BiHalf" means implementing the proposed local match method and the bi-half fine-tuning with the baseline GeM. "SP" refers to the spatial verification re-ranking [87].

the query region of interest. We compare the top 5 retrieval results with the green outline denoting correct retrieval results while red markings denote incorrect results. Same query images are considered as in Figure. 4.6 of Chapter. 4. As we can observe, the proposed local match method also globally outperforms the baseline GeM model, whose retrieved images are shown underneath. The local match method correctly provides all top 5 retrievals for all four query images, while the baseline GeM leads to several incorrect

outcomes.

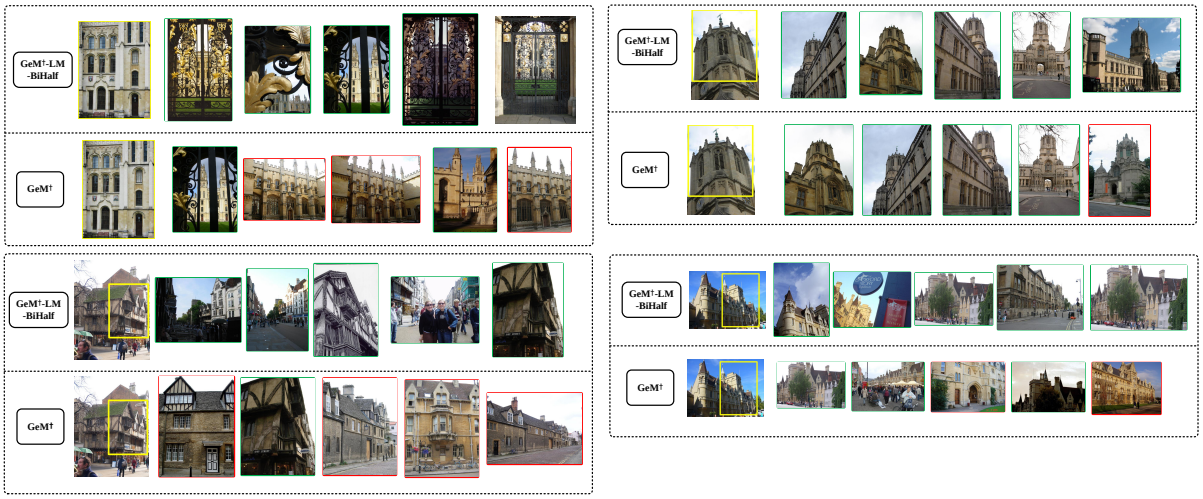


Figure 5.5: Top 5 retrieval results for the proposed Local Match method (with PCA dimension reduction and Bi-half fine-tuning applied) and GeM on images from ROxf dataset.

5.5 Ablation experiment and discussion

In this section, we provide more detailed ablation studies about the impact of each module and the hyper-parameters on retrieval performance.

5.5.1 Binarization and Bi-half fine-tuning impact

We first verify the impact of the Bi-hash fine-tuning on the model's performance. The first row from Table 5.2 shows the results when applying the proposed local match method with the baseline GeM but without binarization, PCA dimension reduction and Bi-half fine-tuning. In this case, all similarity scores are calculated with cosine similarity. The second row in Table 5.2 provides the results when applying the binarization but without considering PCA dimension reduction and Bi-half fine-tuning. Clustered local features $\mathbf{X}_{q,K}$ and $\mathbf{X}_{c,N}$ are both directly binarized by the Sign function Eq. (5.1) then perform the local match as described in Section 5.3.3. The third row applies PCA dimension reduction, reducing the feature dimension to $D_B = 512$ without the Bi-half fine-tuning. The fourth row represents the intact pipeline of the local match method with PCA dimension

Binary	Bi-half fine-tune	PCA	Feature dimension	*Medium (%)*				*Hard (%)*			
				ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
✗	✗	✗	2048	86.4	77.3	92.3	80.0	71.5	55.4	84.1	62.0
✔	✗	✗	2048	86.1	76.7	92.2	79.3	71.2	54.6	84.0	61.1
✔	✗	✔	512	85.4	75.3	90.9	78.0	70.4	52.6	82.3	59.2
✔	✔	✔	512	86.7	76.6	92.0	79.3	72.0	54.8	83.6	61.4

Table 5.2: Ablation experimental results when considering the Bi-half fine-tuning. The column "PCA" indicates whether PCA is applied for dimension reduction. The column "Bi-half fine-tune" indicates whether the whitening layer and PCA parameters are fine-tuned, as illustrated in Figure. 5.1 (c). "Feature dimension" provides the dimension of the final output feature.

reduction and Bi-half fine-tuning both applied. When applying both PCA dimension reduction and binarization but without Bi-half fine-tuning, we can observe that the model performance will decrease significantly, especially considering the 1 million distractor set. After Bi-half fine-tuning, the performance deterioration is greatly relieved. The mAP on the *Hard* set of ROxf even increases from 70.4% to 72.0%.

5.5.2 Different way of Bi-half layer implementation

The original Bi-half layer from [73] is applied on each batch feature at the training stage, while we use the Bi-half layer on class proxy features from the ArcFace loss function Eq. (4.4) at the fine-tuning stage. Here, we also explore the effect caused by these two different ways of Bi-half layer implementation. Table 5.3 presents retrieval results with different Bi-half layer implementation. We can observe that, after 60 epochs of training (fine-tuning) in total, applying the Bi-half layer on proxy features leads to better retrieval results. In addition, according to our observation, applying the Bi-half layer on each batch feature at the training stage makes the model converge slower than when applying it on the proxy features. As discussed in Section 5.3.2, directly applying the Bi-half layer to the batch feature could cause an unstable feature value assignment. On the contrary, applying it to the proxy feature could optimise the model towards a consistent goal across all batch steps, eliminating the distraction caused by the batch size setting or the random image sample shuffle during the training stage.

Bi-half	*Medium* (%)		*Hard* (%)	
apply	ROxf	RPar	ROxf	RPar
Batch	85.6	91.3	71.1	82.8
Proxy	86.7	92.0	72.0	83.6

Table 5.3: Retrieval results on ROxf and RPar datasets when considering different ways for the Bi-half layer implementation. "Proxy" means that it is applied on proxy features from ArcFace loss, as described in Section 5.3.2, "Batch" means it is applied on each batch of the feature at the training stage, as described in [73]. Both models are trained (fine-tuned) according to the description from Section 5.4.1 with no more than 60 epochs.

5.5.3 Why implementing the Bi-half layer at the fine-tuning stage

The dimension reduction is performed by PCA after the baseline model training stage, as shown in Figure 5.1 (b), is finished. Then, the whitening layer along with the PCA parameters: $\mathbf{m}_{\mathcal{W}}$, $\mathbf{P}_{\mathcal{W}}$ are separately fine-tuned as shown in Figure 5.1 (c). One concern could be that why not directly train the whole pipeline in Figure 5.1 (c) in an end-to-end manner from scratch[10]. There are two main reasons accounting for this: First, sharing the same backbone but only fine-tuning the whitening layer while the PCA components could give a fairer comparison to the baseline GeM model. In addition, it makes the ablation study of dimension reduction, as discussed in Section 5.5.4 more fair and convenient as the only changes for each dimension setting are to modify the PCA output dimension and then do the fine-tuning for only 10 epochs. Second, separately fine-tuning the whitening layer and PCA could greatly reduce the global training time cost. According to the testing results, with 4 NVIDIA Tesla GPU, for 60 (50+10) epochs of training, optimizing the whole pipeline from Figure 5.1 (c) in an end-to-end manner takes more than 14 days. For comparison, when first optimising the baseline GeM model for 50 epochs and then fine-tune the whitening layer and PCA components for 10 epochs, it will take around 11 days (10 days for baseline GeM, less than 1 day for the fine-tuning). In other words, directly training the whole pipeline will cause extra time costs for training but without any further performance improvement.

[10]Under this circumstance the ArcFace loss proxies are initialized with random values and the PCA dimension module could be replaced by a fully connected layer which is also randomly initialized.

5.5.4 The impact of PCA dimension reduction

Table 5.4 presents the ablation experiment results with respect to varying the feature dimension. We can change the feature vector dimension by modifying hyper-parameter D_B, as explained in Section 5.3.2. Models with different dimension outputs are separately Bi-half fine-tuned as illustrated in the pipeline from Figure. 5.1 (c). Generally, a larger feature dimension space would lead to better retrieval results. However, it would also require extra computation costs. According to Table 5.4, increasing the feature dimension from the default setting of 512 to 1024 brings minimal improvements. Meanwhile, reducing the feature dimension to 256 or even smaller will cause great performance degradation. Accordingly, $D_B = 512$ represents a good balance between performance and computation cost.

Feature Dimension	*Medium* (%)		*Hard* (%)	
	ROxf	RPar	ROxf	RPar
128	83.6	89.6	68.5	80.8
256	86.5	91.3	71.6	82.3
512	86.7	92.0	72.0	83.6
1024	86.9	92.1	72.3	83.8

Table 5.4: Retrieval results on ROxf and RPar datasets when considering PCA dimension reduction.

5.5.5 Impact of scales

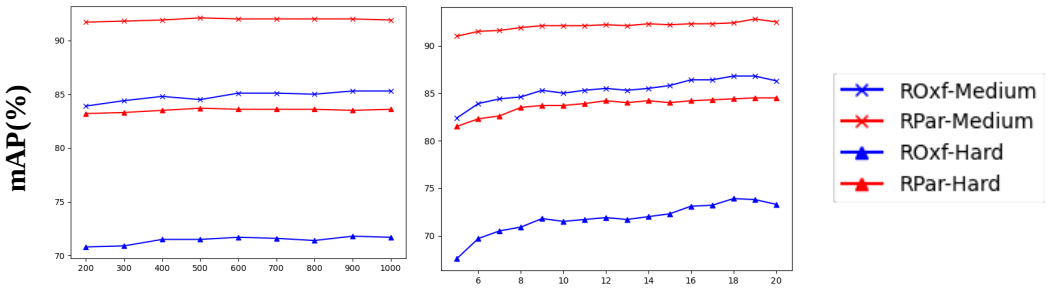
Retrieval results of the proposed method "GeM-LM-BiHalf" with different image scales in Table 5.5. We still consider the three existing scale combinations implemented in the literature : $\left\{\frac{1}{\sqrt{2}}, 1, \sqrt{2}\right\}$ from [98], $\left\{\frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}\right\}$ from [146], and $\left\{\frac{1}{4}, \frac{1}{2\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, 1, \sqrt{2}, 2\right\}$ from [14, 123]. As we can observe from Table 5.5, considering the combination of 5 scales gives the best result for our local match method. Using 7 scales brings little improvement on ROxf dataset but comes with more computation cost for the feature extraction.

1	$\frac{1}{\sqrt{2}}$	$\sqrt{2}$	$\frac{1}{2\sqrt{2}}$	$\frac{1}{2}$	$\frac{1}{4}$	2	*Medium* (%)		*Hard* (%)	
							ROxf	RPar	ROxf	RPar
✔	-	-	-	-	-	-	82.7	88.6	66.0	78.1
✔	✔	✔	-	-	-	-	84.4	90.4	69.0	81.0
✔	✔	✔	✔	✔	-	-	86.7	92.0	72.0	83.6
✔	✔	✔	✔	✔	✔	✔	87.2	92.1	73.0	83.6

Table 5.5: Retrieval results on ROxf and RPar when considering different scales.

5.5.6 Impact of clustering parameters

Two diagrams in Figure. 5.6 (a), (b) show the impact of the number of the initially selected features N as well as the number of clusters K on the retrieval performance. A small $N = 200$ could not cover enough local features, while $N = 1000$ is too large and may pick out too many backgrounds or irrelevant local features, making the feature selection meaningless. A smaller K could further reduce the computation cost, but it would arbitrarily fuse many local features into larger clusters reducing the local matching benefits. A relatively larger K could further improve the retrieval performance as it leads to more detailed clustering, but it would also result in additional computation costs with a marginal improvement.



(a) Ablation study on parameter N (b) Ablation study on parameter K

Figure 5.6: Ablation study on clustering parameters.

5.5.7 Clustering selection

In this ablation study, we present the impact of clustering method selection for the proposed local match pipeline. We still consider two other clustering algorithms: Spectral Clustering [127] and Mean-Shift [22] as described in Section 4.6.3. As we can see from Table 5.6, for the proposed local feature match pipeline, spectral clustering still bring no

Cluster Method	*Medium* (%)		*Hard* (%)	
	ROxf	RPar	ROxf	RPar
Spectral	86.6	92.0	71.9	83.4
k-means++	86.7	92.0	72.0	83.6

Table 5.6: Retrieval results on ROxf and RPar with Spectral Clustering.

Cluster method	Bandwidth	Feature number	*Medium* (%)		*Hard* (%)	
			ROxf	RPar	ROxf	RPar
Mean-Shift	0.5	325	87.0	91.8	72.7	82.7
	1	76	85.2	90.5	70.9	81.8
	1.5	11	83.4	90.0	67.6	81.6

Table 5.7: Retrieval results on ROxf and RPar datasets with Mean-Shift clustering and different bandwidth setting. "feature number" indicates the average number of local features after clustering

advantages but just causes more time cost when compared to the k-means clustering.

According to Table 5.7 Mean-Shift is still outperformed by k-means with the local feature match pipeline. Accordingly, despite its straightforwardness, k-means is still the most effective clustering method.

5.5.8 Impact of query crop

As mentioned before, the standard evaluation protocol of ROxf/RPar dataset requires cropping each query image with the provided bounding box. In this ablation study, we also evaluate the impact caused by the query crop.

Table 5.8 provides the retrieval results of the baseline "GeM†", the proposed method "GeM†-LM-BiHalf" and the current state-of-the-art work DOLG with/without the query image crop. Without the query crop means utilizing more context content for image retrieval, and it significantly improves retrieval accuracy when considering the 1 million distractor set. According to Table 5.8, with or without the query crop, the proposed method "LM-BiHalf" always improves the baseline model's performance, outperforming the current state-of-the-art DOLG on ROxf dataset and giving comparable results on RPar+1M dataset.

5.5.9 Impact of the local match strategy

As described in Section. 5.3.3, for each clustered query local feature $\mathbf{b}_{q,i}$, we define its similarity score $s(\mathbf{b}_{q,i}, \mathbf{I}_c)$ to the candidate image $\mathbf{I}_c$ with the maximum value from the

Method	query crop	*Medium (%)*				*Hard (%)*			
		ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
GeM†	✗	82.5	78.4	90.7	81.3	62.9	56.1	81.0	65.1
GeM†	✓	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
DOLG [146][8]	✗	83.2	79.0	91.6	82.9	64.8	57.9	82.6	67.3
DOLG [146][8]	✓	82.3	73.6	90.9	80.4	64.9	51.6	81.7	62.9
GeM†-LM-BiHalf	✗	86.4	80.8	92.8	81.1	72.1	61.9	84.3	64.2
GeM†-LM-BiHalf	✓	86.7	76.6	92.0	79.3	72.0	54.8	83.6	61.4

Table 5.8: Retrieval results on ROxf and RPar datasets without query crop.

similarity matrix $\mathbf{M}$ (Eq. (5.8)), while the final similarity score $S(\mathbf{I}_q, \mathbf{I}_c)$ between the query image pair $\mathbf{I}_q$ and $\mathbf{I}_c$ is the mean of all query local features' match scores (Eq. (5.9)).

Further ablations are performed first for defining the query local feature's similarity score $s(\mathbf{b}_{q,i}, \mathbf{I}_c)$ from Eq. (5.8) and second, for how to fuse all $s(\mathbf{b}_{q,i}, \mathbf{I}_c)$ into the final image pair similarity score $S(\mathbf{I}_q, \mathbf{I}_c)$. Specifically, we consider 3 different definitions: "Max" means using the maximum value, "Mean" means calculating average value, and "SoftMax" means applying SoftMax function overall values and then performing a weighted sum. The default local match strategy that described in Section. 5.3.3 corresponds to the first row of Table 5.9, in which the definition of $s(\mathbf{b}_{q,i}, \mathbf{I}_c)$ is "Max" while that of $S(\mathbf{I}_q, \mathbf{I}_c)$ is "Mean". As we can observe, the chosen setting gives the best retrieval performance in all these databases.

$s(\mathbf{b}_{q,i}, \mathbf{I}_c)$ define	$S(\mathbf{I}_q, \mathbf{I}_c)$ define	*Medium* (%)		*Hard* (%)	
		ROxf	RPar	ROxf	RPar
Max	Mean	86.7	92.0	72.0	83.6
Max	Max	77.9	88.8	59.7	76.2
Max	SoftMax	86.6	91.9	72.1	83.3
Mean	Max	55.4	77.9	41.0	64.3
Mean	Mean	77.0	87.6	59.6	76.1
Mean	SoftMax	76.8	87.5	59.3	75.8
SoftMax	Max	62.7	81.7	46.5	67.8
SoftMax	Mean	79.8	88.9	62.8	78.1
SoftMax	SoftMax	79.6	88.8	62.7	77.9
baseline GeM		83.0	90.2	65.5	80.7

Table 5.9: Retrieval results on ROxf and RPar when considering different ways to calculate the feature similarity.

5.5.10 Impact of inverted file indexing

As shown in Table 5.10, the local match method "GeM†-LM-BiHalf" gives almost the same mAP results across ROxf/RPar datasets no matter with or without the Inverted File Indexing (IVF). As discussed in Section 4.6.10, the inverted file indexing module is only a coarse-level filter that aims to speed up the online retrieval process by filtering out easy negative images. Therefore, it makes almost no difference to the model's accuracy.

Method	IVF	*Medium (%)*				*Hard (%)*			
		ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
GeM†-LM-BiHalf	✗	86.6	76.6	92.1	79.1	72.0	54.6	83.8	61.0
GeM†-LM-BiHalf	✔	86.7	76.6	92.0	79.3	72.0	54.8	83.6	61.4

Table 5.10: Retrieval results on ROxf and RPar datasets with/without inverted file indexing.

5.5.11 Computation cost

In the following, we discuss the computation and memory costs of the proposed local match method considering the hyper-parameter setting described in Section 5.4.1. After the PCA dimension reduction and binary encoding, each element of the clustered local feature vector is represented as a 1-bit binary number. In this case, for each candidate image, the memory cost to cache its local features is $K \times D_B \times 1$ bits. With PCA output feature dimension $D_B = 512$, $K = 10$, the memory cost for one candidate image cache is $10 \times 512 \times 1$ bit ≈ 0.00064 MB. It takes around 0.64GB to cache the ROxf/RPar dataset along with the +1M distractor set.

The feature extraction, including feeding through the backbone network, feature selection and clustering takes in average 210ms to cache a single candidate image's local features when considering 5 input image scales. This kind of processing becomes time consuming when considering large-scale databases, but it is done offline and only once.

Considering a pair of the query image and candidate image of size 512×512 in original scale as input, ResNet101(R101) [45] as the backbone network for GeM feature extraction, Table 5.11 (a) and (b) compare the time cost of the GeM model at the online retrieval stage with or without the proposed local match method. $K = 10$ clustered binary local

For query image			Local feature
Backbone(R101)	Feat select and clustering	PCA	match (CPU)
16	15	0.04	0.11

(a) Time cost of each component within the proposed local feature match pipeline at the online evaluation stage.

For query image		Cosine
Backbone(R101)	Pool and whiten	Similarity
16	0.2	0.05

(b) Time cost of each component within the GeM [98] pipeline at the online evaluation stage.

Table 5.11: Time cost analysis of each component within the local feature match pipeline (a) and the GeM pipeline (b) at the online evaluation stage. A pair of one query image and one candidate image serves as input. Features of the candidate image have been pre-extracted as the "offline stage" shown in Fig. 1.2. The time cost of each component is reported in milliseconds (ms).

features of the candidate image have been pre-cached at the offline stage. All experiments were performed 100 times, and we report the average time cost of each component in Table 5.11.

As we can observe, the extra time cost of the proposed local match method is caused by two facts: first, for the query feature extraction, local features of the query image need to be selected and clustered along with binarization by the Sign function. Second, the local feature match (with CPU) takes more time than simply performing cosine similarity measures on GPU.

For the online retrieval searching on ROxf/RPar with +1M distractor dataset, with the help of inverted file indexing, for one query image it takes on average 0.59s (without the inverted file indexing it would be around 1.8s) with the python implementation on a CPU. Detailed computation cost requirements and comparison with other models are provided in Table 5.12. Our method "GeM†-LM-BiHalf" requires much less memory requirement with a comparable time cost compared to existing works.

5.5.12 Comparison with co-attentions

As mentioned in Section 5.4, the local match could work like co-attention. This section compares the local match method with the two co-attention works from the former chapters. In Table. 5.13, we quantitatively compare the retrieval performance with the

Method	Device	Memory (GB) ROxf/RPar+1M	Retrieval time (ms) in average
HOW [123]	CPU	14	750
GeM [98]	Tesla GPU	8	250
DOLG [146]	Tesla GPU	2	220
DELG+SP [14]	Tesla GPU	22	383
GeM†-LM-BiHalf (ours)	CPU	0.64	590

Table 5.12: Computation cost comparison.

Conditional Attention Network "CANet" method from Chapter 3, clustering based co-attention "CA(cluster)" from Chapter 4 on ROxf/RPar datasets. As each of the three methods mentioned above could be treated as a post-processing module for pre-trained CNN feature re-weighting or local matching, for fair comparison, the GeM model described in Section 4.2.2 is used as the baseline model. As we can observe, all three methods greatly boost the baseline model's performance. Although the "CA(cluster)" gives the best result, the "LM-BiHalf" has comparable accuracy with much less computation cost.

Method	*Medium (%)*				*Hard (%)*			
	ROxf	ROxf+1M	RPar	RPar+1M	ROxf	ROxf+1M	RPar	RPar+1M
R101-GeM†(baseline)	83.0	72.8	90.2	77.6	65.5	49.8	80.7	59.1
R101-GeM†-CANet	84.3	74.5	91.0	78.7	68.9	51.4	82.0	60.8
R101-GeM†-CA(cluster)	86.4	79.3	93.2	81.8	72.6	59.9	85.6	64.1
R101-GeM†-LM-BiHalf	86.7	76.6	92.0	79.3	72.0	54.8	83.6	61.4

Table 5.13: Retrieval results on ROxf and RPar datasets. "R101-GeM†(baseline)" indicate the baseline model as described in Section 4.2.2. "R101-GeM†-CANet" represents the baseline model combined with CANet from Chapter 3. "R101-GeM†-CA(cluster)" represents the baseline model combined with the clustering-based co-attention method from Chapter 4. And the "R101-GeM†-LM-BiHalf" represents the local match method proposed in this chapter.

In Figure 5.7, we compare the generated attention map between the three works. All three methods could generate good query sensitive attention maps. Due to the usage of convolution layer based fusion module, the CANet tends to highlight a regular square area. The clustering-based co-attention and the local match are both based on local feature clustering. As a result, highlighted regions could be irregularly shaped. Especially, limited by the computation cost, the cluster count is manually set to a small value ($K = 10$), and some neighbour locations that belong to different objects may inevitably be grouped together, making the final attention map not accurate enough. As the example 3 in Figure 5.7, the region of the target building along with some spire structures at the

bottom-left side, which is quite similar to the top parts of the target building, are equally highlighted. On the contrary, the CANet gives a better attention map even though it is quantitatively outperformed with respect to global retrieval accuracy, as shown in Table. 5.13.

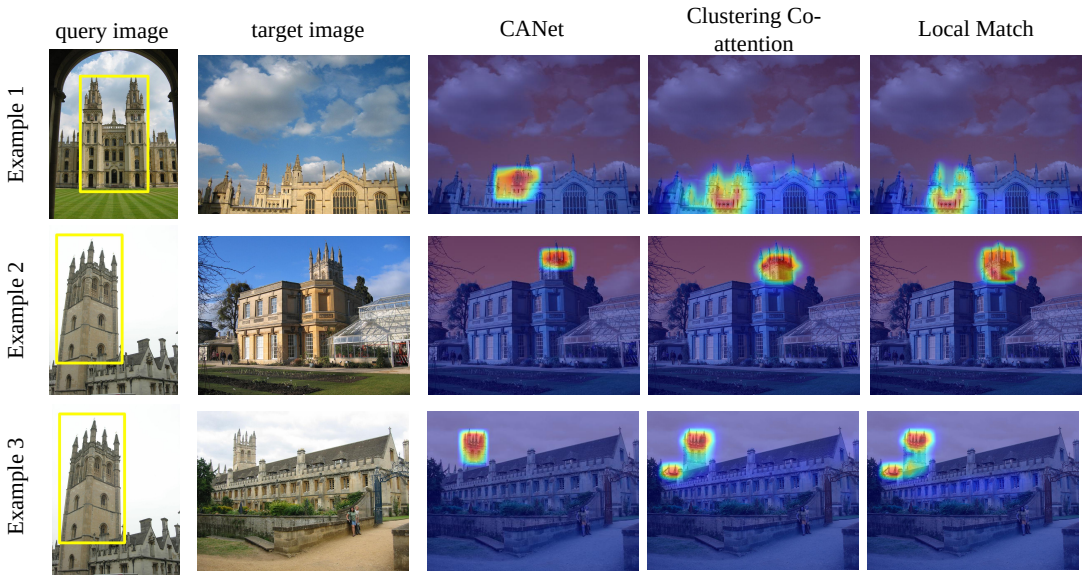


Figure 5.7: Attention map visualization. The first column shows the query image with a yellow bounding box outlining the target object. The second column is the target image. The third column represents the co-attention map generated by CANet, the forth column shows the co-attention map generated by the co-attention method from 4 and the final column shows the local match that was generated as description from Section 5.3.3.

5.6 Conclusion

In this chapter, we explore performing the many-to-many local matching with extracted few but expressive clustered local features from images. Unlike other local matching methods, which extract large numbers of low-dimensional local features and may require complex match kernel implementation, the proposed local matching method is simple but effective when applied with clustered local features. The usage of PCA dimension reduction and Sign function based binarization significantly decreases the computation costs. While the adapted Bi-half layer fine-tuning procedure enriches the information capacity of each feature channel, relieving the information loss problem caused by feature compression. With the proposed CBIR pipeline, the method achieves new state-of-the-art performances on benchmark datasets with much lower memory costs than existing methods.

Some interesting conclusions could be drawn from the experimental results of the local matching method in this chapter. First, as demonstrated in Figure 5.4, the local match using those few but expressive clustered local features could lead to a co-attention-like match map. The resulting match map is also query sensitive, varying the effectiveness and contribution of each candidate image local feature with the actual input query image local features. Compared to co-attention based global feature re-weighting, the local feature match method could cooperate with binary encoding, which leads to much lower computation costs.

Second, at the fine-tuning stage, enforcing the proxy feature from the ArcFace loss function has half-half binary value distribution, instead of enforcing each batch feature has that, could lead to a more stable training procedure and still enrich the information capacity of each feature channel, relieving the performance degradation problem caused by dimension reduction and binarization.

Chapter 6

Conclusion

In this thesis, we explored introducing query sensitive attention mechanisms for content-based image retrieval. Different ways of co-attention generation or local feature matching are proposed to boost retrieval performance. The first work: Conditional Attention Network (CANet), serves as a separate trainable co-attention generation branch for each candidate image. It is trained under the supervision of the SuperPoint model, a self-supervised match point detector for image pairs. The CANet generates co-attention based on stacks of convolution layers with different kernel sizes. Although it positively impacts the final retrieval accuracy, it comes with unbearable extra computation costs, making it impractical for large-scale retrieval tasks. Then, a more straightforward non-trainable co-attention generation method is proposed. It is based on local feature clustering and serves as a post-processing module for the pre-trained CNN feature output. The clustering-based co-attention method dramatically improves the baseline GeM model's retrieval accuracy, reaching new state-of-the-art results on benchmark datasets with comparable computation costs to existing works. Instead of trying to extract a co-attention weighted global feature vector as the former two works, the third work: expressive local feature matching, employs clustering in the feature space for efficiently extracting characteristic features for image retrieval. The proposed local match method works with binary encoding for further feature compression. Meanwhile, the Bi-half layer based fine-tuning procedure greatly relieves the information loss caused by dimension reduction and binarization. The expressive local feature matching method shows comparable retrieval results to the former clustering-based co-attention but with much less computation cost and gets rid of reliance

on GPU at the online retrieval stage. In addition, although the local feature matching method does not explicitly generate a co-attention map for feature re-weighting as the former two works, according to the visualization results, it implicitly leads to query-sensitive local match maps, which work like co-attention.

According to the experimental results of the proposed works, as mentioned above, we can draw the following conclusions for query-sensitive attention enabled CBIR framework. First, embedding the co-attention mechanism into the feature extraction pipeline can significantly improve retrieval accuracy, leading to new state-of-the-art retrieval results on current benchmark datasets ROxf/RPar datasets. Considering the interaction between the query global feature to candidate local features output by CNN is an intuitive and effective way for co-attention generation. After well-training, clustering over local features from convolution feature tensor could automatically group local features that belong to the same object together, resulting in few but expressive clustered local feature representations for the input image. Finally, the local feature matching could be treated as an imitation of co-attention, implicitly resulting in query-sensitive co-attention-like local match maps. After combining with dimension reduction and binarization methodology, the local feature match method could give comparable retrieval accuracy to the co-attention method but comes with much less computation cost at the retrieval stage.

6.1 Future work

In this section, we discuss possible future development in the field of content-based image retrieval.

6.1.1 Better backbone network structure

All proposed works from the former chapters utilize a fully convolutional network as the backbone structure. Although it already gives good retrieval performance, the recently proposed Vision Transformer (ViT) [32, 28, 76] has demonstrated superior feature extraction capability in computer vision tasks. The ViT is based on Self-Attention [126], which enables the feature extraction pipeline with a global receptive field and dynamic

weighting to improve the model's feature extraction ability. These properties lead to several advantages such as higher model capacity [28], better ability to capture long-range dependencies, complex interactions between positions and ability to process high-level concepts during image recognition [84]. Replacing the CNN with ViT could be a promising direction for further retrieval performance improvement. With the advantages of ViT mentioned above, it should be able to extract more comprehensive local features, leading to better attention maps and global retrieval performance.

6.1.2 Jointly trainable cluster

The clustering-based local feature extraction strategy, used in both Chapter 4 and Chapter 5, serves as a non-trainable post-processing feature extraction module for feature tensor output by pre-trained CNN. In other words, the local feature from each entry on the convolution feature tensor is only implicitly optimized with a global loss at the pre-training stage, while the actual utilization of these local features is in a local match (global-to-local or local-to-local) manner without any fine-tuning or optimization. This discrepancy could lead to sub-optimal feature extraction and global model performance. Considering an end-to-end trainable feature clustering module and optimizing local features in a more specific way could be another good research direction for better retrieval performance.

Bibliography

[1] R Achanta, S Hemami, F Estrada, and S Susstrunk. Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1597–1604, 2009.

[2] Swati Agarwal, Anil Kumar Verma, and Preetvanti Singh. Content based image retrieval using discrete wavelet transform and edge histogram descriptor. In *Proceedings of the International Conference on Information Systems and Computer Networks*, pages 19–23, 2013.

[3] Ahmad Alzu'bi, Abbes Amira, and Naeem Ramzan. Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation*, 32:20–54, 2015.

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[5] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 8, pages 1027–1035, 2007.

[6] Yannis Avrithis and Giorgos Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International journal of computer vision*, 107(1):1–19, 2014.

[7] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015.

[8] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 584–599, 2014.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[10] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[11] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[12] Albrecht Blaser. *Data Base Techniques for Pictorial Applications*. Springer, 1980.

[13] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 677–694. Springer, 2020.

[14] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 726–743, 2020.

[15] Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang. Edgel index for large-scale sketch-based image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–768, 2011.

[16] Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. Mindfinder: interactive sketch-based image search on millions of images. In *Proceedings of ACM International Conference on Multimedia (MM)*, pages 1605–1608, 2010.

[17] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5608–5617, 2017.

[18] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.

[20] Wei Chen, Yu Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep learning for instance retrieval: A survey. *arXiv preprint arXiv:2101.11282*, 2021.

[21] Yudong Chen, Zhihui Lai, Yujuan Ding, Kaiyi Lin, and Wai Keung Wong. Deep supervised hashing with anchor graph. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9796–9804, 2019.

[22] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(8):790–799, 1995.

[23] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1258, 2017.

[24] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 539–546, 2005.

[25] Ondřej Chum, Andrej Mikulik, Michal Perdoch, and Jiří Matas. Total recall ii: Query expansion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–896, 2011.

[26] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[27] Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. Stochastic generative hashing. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 913–922, 2017.

[28] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[29] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019.

[30] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 224–236, 2018.

[31] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi, Yanhua Yang, Cheng Deng, and Heng Huang. Unsupervised deep generative adversarial hashing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3664–3673, 2018.

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[33] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[34] Noa Garcia and George Vogiatzis. Learning non-metric visual similarity for image retrieval. *Image and Vision Computing*, 82:18–25, 2019.

[35] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[36] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[37] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[38] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[39] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 241–257, 2016.

[40] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. End-to-End learning of deep visual representations for image retrieval. *International Journal of Computer Vision (IJCV)*, 124(2):237–254, 2017.

[41] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 545–552, 2007.

[42] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[45] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[46] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2725–2734, 2019.

[47] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[48] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 762–768, 1997.

[49] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of IEEE computer society conference on Computer Vision and Pattern Recognition*, pages 762–768. IEEE, 1997.

[50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[51] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2077–2086, July 2017.

[52] L Itti, C Koch, and E Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(11):1254–1259, 1998.

[53] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[54] Young Kyun Jang and Nam Ik Cho. Self-supervised product quantization for deep unsupervised image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12085–12094, 2021.

[55] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–317. Springer, 2008.

[56] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[57] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010.

[58] Qing-Yuan Jiang and Wu-Jun Li. Asymmetric deep supervised hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[59] HeeJae Jun, Byungsoo Ko, Youngjoon Kim, Insik Kim, and Jongtack Kim. Combination of multiple global descriptors for image retrieval. *arXiv preprint arXiv:1903.10663*, 2019.

[60] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Proceedings of the European Conference on Computer Vision workshops*, pages 685–701, 2016.

[61] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 506–513, 2004.

[62] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR), arXiv preprint arxiv:1412.6980*, 2015.

[63] Alex Krizhevsky and Geoffrey E Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, volume 1, page 2. Citeseer, 2011.

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 25:1097–1105, 2012.

[65] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2130–2137. IEEE, 2009.

[66] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3270–3278, 2015.

[67] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5639–5650, 2020.

[68] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[69] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[70] Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, pages 261–276, 1995.

[71] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[72] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees GM Snoek, and Alberto Del Bimbo. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys (CSUR)*, 49(1):1–39, 2016.

[73] Yunqiang Li and Jan van Gemert. Deep unsupervised image hashing by maximizing bit entropy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2002–2010, 2021.

[74] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 27–35, 2015.

[75] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2064–2072, 2016.

[76] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[77] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[78] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(8):837–842, 1996.

[79] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 287–304, September 2018.

[80] Eva Mohedano, Kevin McGuinness, Xavier Giro-i Nieto, and Noel E. O'Connor. Saliency weighted convolutional features for instance search. In *Proceedings of the International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2018.

[81] Eva Mohedano, Kevin McGuinness, Noel E O'Connor, Amaia Salvador, Ferran Marques, and Xavier Giró-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, pages 327–331, 2016.

[82] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 360–368, 2017.

[83] B. Munjal, S. Amin, F. Tombari, and F. Galasso. Query-guided end-to-end person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 811–820, 2019.

[84] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[85] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. Solar: second-order loss and attention for image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 253–270, 2020.

[86] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, 2006.

[87] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3456–3465, 2017.

[88] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. SalGAN: Visual saliency prediction with generative adversarial networks. In *arXiv preprint arxiv:1701.01081*, 2017.

[89] Alex Papushoy and Adrian G. Bors. Image retrieval based on query by saliency content. *Digital Signal Processing*, 36:156–173, 2015.

[90] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM workshops on Multimedia*, pages 51–54, 2000.

[91] Greg Pass and Ramin Zabih. Histogram refinement for content-based image retrieval. In *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV)*, pages 96–102, 1996.

[92] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391. IEEE, 2010.

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

[94] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[95] Guoping Qiu. Color image indexing using btc. *IEEE transactions on image processing*, 12(1):93–101, 2003.

[96] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5706–5715, 2018.

[97] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20. Springer, 2016.

[98] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(7):1655–1668, 2018.

[99] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

[100] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6):1137–1149, 2016.

[101] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5107–5116, 2019.

[102] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision (IJCV)*, 40(2):99–121, 2000.

[103] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[104] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, 105(3):222–245, 2013.

[105] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[106] Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Spatially-constrained similarity measurefor large-scale object retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1229–1241, 2013.

[107] Yuming Shen, Li Liu, and Ling Shao. Unsupervised binary representation learning with deep variational networks. *International Journal of Computer Vision (IJCV)*, 127(11):1614–1628, 2019.

[108] Yuming Shen, Jie Qin, Jiaxin Chen, Mengyang Yu, Li Liu, Fan Zhu, Fumin Shen, and Ling Shao. Auto-encoding twin-bottleneck hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2827, 2020.

[109] Christian Siagian and Laurent Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):300–312, 2007.

[110] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11651–11660, 2019.

[111] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1409.1556*, 2015.

[112] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 3, pages 1470–1470, 2003.

[113] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen. Binary generative adversarial networks for image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 394–401, 2018.

[114] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 5551–5560, 2017.

[115] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[116] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[117] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision (IJCV)*, 7(1):11–32, 1991.

[118] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[119] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[120] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5109–5118, 2019.

[121] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision (IJCV)*, 116(3):247–261, 2016.

[122] Giorgos Tolias and Hervé Jégou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern recognition*, 47(10):3466–3476, 2014.

[123] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 460–477, 2020.

[124] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *Proceedings of the International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1511.05879*, pages 1–12, 2016.

[125] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[126] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[127] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[128] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166, 2014.

[129] Qi Wang, Jinxiang Lai, Zhenguo Yang, Kai Xu, Peipei Kan, Wenyin Liu, and Liang Lei. Improving cross-dimensional weighting pooling with multi-scale feature fusion for image retrieval. *Neurocomputing*, 363:17–26, 2019.

[130] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1328–1338, 2019.

[131] Shuang Wang and Shuqiang Jiang. Instre: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(3):1–21, 2015.

[132] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[133] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5207–5216, 2019.

[134] Xiang-Yang Wang, Yong-Jian Yu, and Hong-Ying Yang. An effective image retrieval scheme using color, texture and shape features. *Computer Standards & Interfaces*, 33(1):59–68, 2011.

[135] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in Neural Information Processing Systems (NeurIPS)*, 21, 2008.

[136] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2575–2584, 2020.

[137] Hui Wu, Min Wang, Wengang Zhou, and Houqiang Li. Learning deep local features with multiple dynamic attentions for large-scale image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 11416–11425, 2021.

[138] Xiaomeng Wu, Go Irie, Kaoru Hiramatsu, and Kunio Kashino. Weighted generalized mean pooling for deep image retrieval. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 495–499, 2018.

[139] Xuanlu Xiang, Zhipeng Wang, Zhicheng Zhao, and Fei Su. Multiple saliency and channel sensitivity network for aggregated convolutional feature. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9013–9020, 2019.

[140] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[141] Bin Xu, Jiajun Bu, Yue Lin, Chun Chen, Xiaofei He, and Deng Cai. Harmonious hashing. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[142] Hao Xu, Jingdong Wang, Xian-Sheng Hua, and Shipeng Li. Image search by concept map. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 275–282. Association for Computing Machinery, 2010.

[143] Jian Xu, Cunzhao Shi, Chengzuo Qi, Chunheng Wang, and Baihua Xiao. Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[144] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015.

[145] Fei Yang, Jia Li, Shikui Wei, Qinjie Zheng, Ting Liu, and Yao Zhao. Two-stream attentive CNNs for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 1513–1521, 2017.

[146] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. DOLG: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 11772–11781, 2021.

[147] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29, 2016.

[148] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[149] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 685–701, 2015.

[150] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015.

[151] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014.

[152] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32–32, 2008.

[153] Shiliang Zhang, Qi Tian, Ke Lu, Qingming Huang, and Wen Gao. Edge-sift: Discriminative binary descriptor for scalable partial-duplicate mobile search. *IEEE Transactions on Image Processing*, 22(7):2889–2902, 2013.

[154] Liang Zheng, Shengjin Wang, Ziqiong Liu, and Qi Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1947–1954, 2014.

[155] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.

[156] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017.

[157] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 593–602, 2019.

[158] Maciej Zieba, Piotr Semberecki, Tarek El-Gaaly, and Tomasz Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. *Advances in Neural Information Processing Systems*, 31, 2018.

[159] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Computing Surveys (CSUR)*, 38(2):6–es, 2006.