TOWARDS LARGE-SCALE AND FINE-GRAINED IMAGE RECOGNITION

by

Xiaofan Zhang

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2017

Approved by:

Shaoting Zhang

Min Shin

Jianping Fan

Xinghua Shi

ABSTRACT

XIAOFAN ZHANG. Towards large-scale and fine-grained image recognition. (Under the direction of SHAOTING ZHANG)

In this dissertation, we aim to investigate the problem of large-scale and fine-grained image recognition, which focuses on the differentiation of subtle differences among subordinate classes and a large number of images. Particularly, we tackle this problem by answering three inter-related questions: 1) how to learn robust and invariant feature representations that can differentiate subtle and fine-grained differences among subordinate classes, 2) how to index these features for efficient image analysis (e.g., classification, content-based retrieval) at a large scale, and 3) how to fuse different type of features to get better results. We propose a series of methods to solve these three problems. Regarding feature representation learning, we design an architecture of convolutional neural networks (CNNs), by unifying the classification constraint and the similarity constraint in a multi-learning framework. Also, structured labels are embedded in this framework, so the similarity of images can be defined at different levels of relevance, e.g., the number of shared attributes, through learned feature representations. Regarding feature indexing, we propose multiple methods based on hashing and binary coding, enabling real-time image retrieval and classification for high-dimensional features and/or a large number of features. Regarding feature fusion, we employ a graph-based query-specific fusion approach where multiple retrieval results (i.e., rank lists) are integrated and reordered based on a fused graph. We have evaluated these methods on both natural images and medical images, as we advocate that medical image recognition (e.g., cancer grading by histopathological images) needs ultra-fine-grained differentiation. The experimental results demonstrate the efficacy of our methods, in terms of both accuracy and efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Recent efforts (in both academia and industry) in machine learning and computer vision, particularly the convolutional neural networks (CNNs) [45, 53, 100, 95], have led to large-scale, data-driven methods for robust tagging [40, 36, 31], object classification [17, 91, 48], and semantic segmentation [27, 33, 78, 69]. Such "Internet scale" algorithms have been adapted and applied to the problem of fine-grained image recognition, which focuses on the differentiation of subtle differences among subordinate classes, such as different models of cars [50, 52, 65, 124], breeds of animals [49, 82, 18, 5, 51, 63], types of food dishes [7, 126], and even different stages of cancer [29, 88, 106], which could be ultra-fine-grained. Compared to generic image recognition, fine-grained tasks have huge potentials to be applied in practical applications, such as image-based recommendation system in e-commerce (e.g., given a picture of a food dish, one can return a set of dishes with similar flavor), city landmark localization and recognition for tourism, face recognition and/or verification for security requirements, and computer-aided diagnosis (e.g., cancer grading by thoroughly analyzing histopathological images). However, this task also has several main challenges: 1) Many fine-grained classes are highly correlated and are difficult to distinguish due to their subtle differences, i.e., small inter-class variance. 2) On the other hand, the intra-class variance can be large, partially due to different poses and/or viewpoints. This is particularly true in medical image analysis, since differentiating stages of diseases may rely on a thorough examination of subtle changes in local regions. Traditional methods often cannot discriminate among semantically different, but visually similar (or vice-versa) medical images. Fig. 1.1 shows examples of generic, fine-grained and ultra-fine-grained image recognition, illustrating the

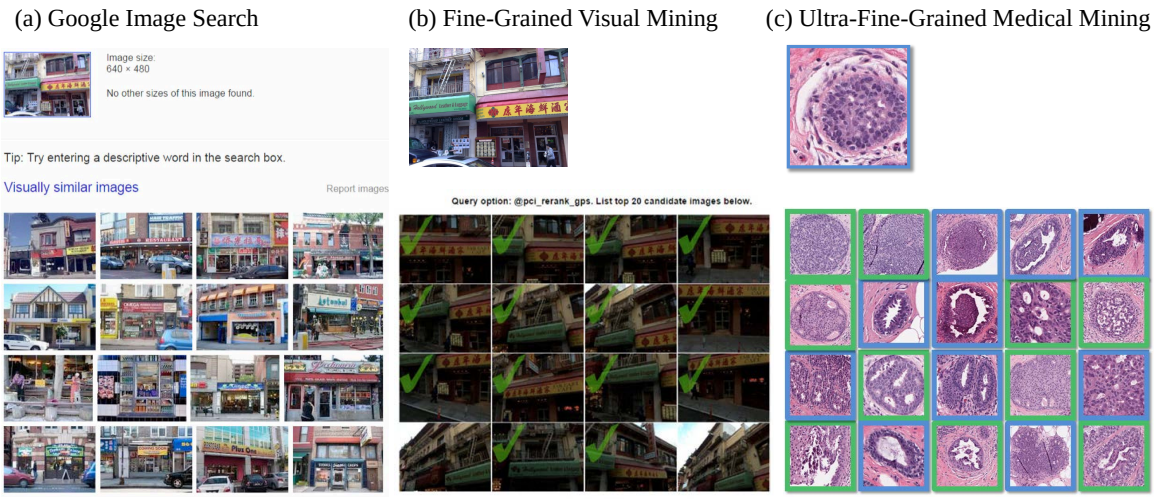


Figure 1.1: Methods designed for generic image retrieval are not well-suited to the challenges of fine-grained and ultra-fine-grained image recognition. For example, (a) a Google Image Search for a San Francisco restaurant returns buildings, but not the same as the query. (b)Recent work for fine-grained indexing of millions of images returns more relevant matches, even with occlusions and viewpoint or illumination changes [133, 134]. It is much better than the approaches for generic image recognition. However, for a query over a large database of histopathological images for breast cancer diagnosis, the results show a mix of images from benign (blue) and actionable (green) cases, even though the query is benign.

challenges of this task and limitations of existing methods. Therefore, in the current era of image recognition, there is an urgent need to improve the performance of fine-grained and even ultra-fine-grained image recognition.

Another important requirement of image recognition in the current scenario is the scalability, i.e., the ability to conduct large-scale image analysis with high efficiency. Take content-based image retrieval (CBIR) as an example, and use medical image analysis as the use case, traditional CBIR methods in this field usually focus on small data sets that have only tens or hundreds of images. New opportunities and challenges arise with the ever-increasing amount of patient data in the current era. Intuitively, larger databases provide more comprehensive information and may improve the accuracy of CBIR systems. On the other hand, achieving an acceptable retrieval efficiency is a challenging task for large-scale data, especially when very large numbers of features are required to capture subtle image descriptors. In fact, CBIR methods

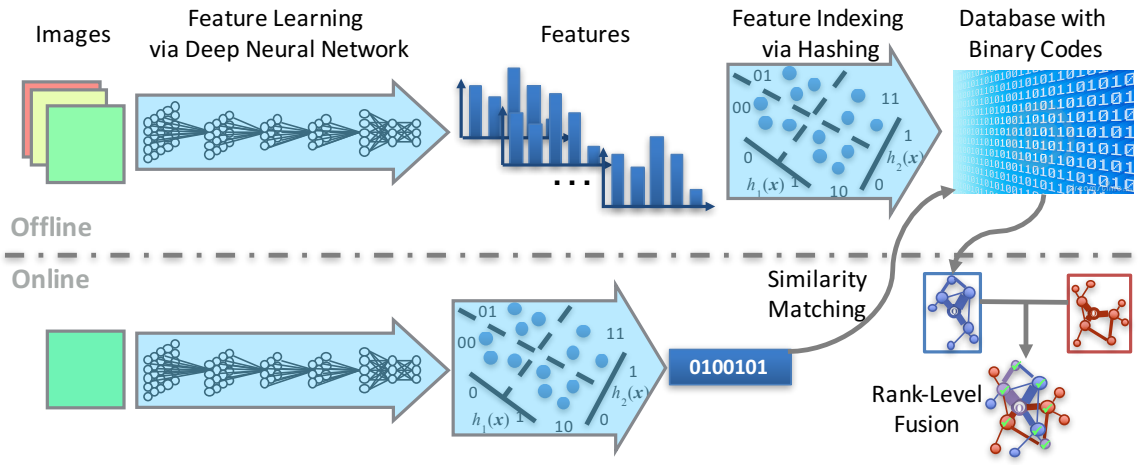


Figure 1.2: Framework of large-scale and fine-grained image recognition, through feature representation learning and indexing. Based on deep neural networks, we learn effective feature representations to differentiate fine-grained differences among images. For scalable analysis, we index a large number of features, either learned or hand-crafted, through hashing methods as binary codes. Then, we could fuse multiple features on the rank-level for better results. Therefore, image recognition tasks such as classification and retrieval can be achieved efficiently.

usually suffer from the "curse of dimensionality" and low computational efficiency when using high-dimensional features in large databases. Although cloud and grid computing are a potential solution for efficient computing [122, 30], few efforts have been made to develop computational and scalable algorithms for large-scale medical image analysis, which is still an urgent need. For more accurate results, we could try to fuse the results of different types of information, e.g. local and holistic feature of images. However, their characteristics, algorithmic procedures and representations can be dramatically different, making them nontrivial to fuse.

To tackle these challenging problems and to achieve fine-grained and large-scale image recognition for various applications including medical images, we conduct research on two inter-related components: 1) learning feature representations for fine-grained differentiation, 2) indexing them for scalable image analysis, and 3) fuse multiple features in rank level, as illustrated in Fig. 1.2. Feature learning or representation learning is a set of techniques that transform raw data input (e.g., images) to a repre-

sentation (e.g., vectors of values) that can be effectively exploited in image recognition tasks, such as classification and retrieval. This is essentially an initial and important procedure for these tasks, since without effective representations, most machine learning methods cannot be applied on fine-grained image databases. To learn such feature representations, we propose methods based on CNNs, by embedding label structures [143]. More specifically, 1) A multitask learning framework is designed to effectively learn fine-grained feature representations by jointly optimizing both classification and similarity constraints in CNNs. 2) To model the multi-level relevance, label structures such as hierarchy or shared attributes are seamlessly embedded into the framework by generalizing the triplet loss in CNNs. We have achieved state-of-the-art performance on four fine-grained datasets. More importantly, it significantly outperforms previous fine-grained feature representations for image retrieval at different levels of relevance. 3) In order to learn features that are invariant to the categorization irrelevant factors (such as pose, viewpoint), triplet loss could be modified in an orthogonal way with those factors information.

Once we have obtained effective feature representations for fine-grained recognition, the remaining issue is to achieve efficient performance of image recognition even when dealing with a large-scale dataset, i.e., scalability. As mentioned above, in this work, we focus on CBIR methods, so we aim to achieve real-time retrieval among large image databases. To this end, we propose a series of feature indexing methods based on hashing and binary coding algorithms, which represent high-dimensional features as tens of bits, without sacrificing their ability of differentiating fine-grained differences [138, 137, 141, 139, 140]. Specifically, we design two hashing approaches, the supervised hashing method and the weighted hashing method, to resolve the limitations of previous work. In addition, we propose to use a graph-based method to fuse multiple features in the rank level to boost the performance further [136, 135]. To validate these proposed methods, we choose histopathological image analysis as

the use case, which is a case of the ultra-fine-grained image recognition. Using their binary codes from hashing functions, we conduct real-time image retrieval among a large number of high-resolution histopathological images, and they can be used to differentiate cancer types or stages.

To summarize, our research work for large-scale and fine-grained image recognition has the following contributions, which will be elaborated in each chapter:

1. We propose robust feature representation learning approaches, which could learn invariant feature and embeds label structures (i.e., relevance at multiple levels) in a multi-task CNN framework. Such fine-grained feature representation can differentiate subtle differences of similar images, at multiple scales of relevance [143].

2. We introduce a supervised hashing method to index high-dimensional feature representation, enabling real-time image retrieval when dealing with large image databases. The supervision also helps to bridge the semantic gap [138, 137].

3. To improve hashing methods for feature indexing, we propose a carefully designed learning method that assigns probabilistic-based importance to different hash values or entries. This scheme alleviates several intrinsic problems of using traditional hashing methods for classification, and significantly improves the accuracy [139, 140, 141].

4. We describe a graph-based framework to fuse the holistic architecture feature and the local appearance feature [136, 135].

5. In addition to the evaluation on natural images, we have also applied our methods to solve a challenging and significant problem, differentiation of cancer types using histopathological images. Particularly, we have conducted cell-level analysis through large-scale image retrieval, by examining half million of cells in real-time, and achieved promising accuracy with thorough experiments [142].

The rest of this dissertation is organized as follows: We first provide a review of related work. Then, we elaborate our contributions on learning fine-grained feature representations in CNNs. In the following chapters, we introduce details of the supervised hashing method for feature indexing and real-time retrieval, with an application of the image-guided diagnosis of intraductal breast lesions using histopathological images. We also propose a weighted hashing method that alleviates the issues of previous work. It is able to analyze millions of cells in histopathological images in real-time, using image retrieval. Further more, we introduce a rank level fusion method to boost the performance. Finally, we conclude this dissertation and provide several potential directions for future work.

CHAPTER 2: RELATED WORK

In this chapter, we review related work in different fields, including: 1) fine-grained image recognition and feature representation learning, mainly based on deep learning approaches, 2) large-scale feature indexing and retrieval by hashing and binary coding methods, and 3) content-based medical image retrieval. We also discuss the limitations of current methods, and elaborate the difference and motivation of our proposed methods.

2.1 Fine-Grained Image Recognition

Fine-grained image understanding aims to differentiate subordinate classes. In this section, we emphasize on the methods that are most relevant to our approaches, particularly the ones on fine-grained feature representation.

Many algorithms have been proposed to leverage parts of objects to improve the classification accuracy. Part based models [125, 10, 4, 130, 129, 34] are proposed to capture the subtle appearance differences in specific object parts and reduce the variance caused by different poses or viewpoints. For example, [117, 64] proposed to combine the part-level and object-level information together to boost the performance. Different from these part-based methods, distance metric learning can also addresses these challenges by learning an embedding such that data points from the same class are clustered together, while those from different classes are pushed apart from each other. In addition, it ensures the flexibility of grouping the same category, such that only a portion of the neighbors from the same class need to be pulled together. For example, Qian et al. [85] proposed a multi-stage metric learning framework that can be applied in large-scale high-dimensional data with high effi-

ciency. In addition to directly classify the images using CNN, it is also possible to generate discriminative features that can be used for classification. In this context, DeCAF [20] is a commonly used feature representation with promising performance achieved by training a deep convolutional architecture on an auxiliary large labeled object database. These features are from the last few fully connected layers of CNN, which have sufficient generalization capacity to perform semantic discrimination tasks using classifiers, reliably outperforming traditional hand-engineered features.

One limitation of the above mentioned methods is that they are essentially driven by the fine-grained class labels for classification, while it is desired to incorporate similarity constraints as well. Therefore, other than using classification constraints alone (e.g., softmax), several similarity constraints have been proposed for feature representation learning. For example, siamese network [13] defines similar and dissimilar image pairs, with the requirement that the distance between dissimilar pairs should be larger than a certain margin, while the one from similar pairs should be smaller. This type of similarity constraint can effectively learn feature representations for various tasks, especially for the verification [116, 93]. An intuitive improvement is to combine the classification and the similarity constraints together for better performance. This is particularly relevant to our framework. For example, [99, 127] proposed to combine the softmax and contrastive loss in CNN via joint optimization. It improved traditional CNN because contrastive constraints might augment the information for training the network.

Different from these approaches, our method leverages the triplet constraint [80, 11] instead of the contrastive ones, since triplet can preserve the intra-class variation [90], which is critical to the learning of fine-grained feature representation. Note that triplet constraint has been used in feature learning [111, 58, 105], face representation [90], and person re-identification [19]. Particularly, there are also efforts on combining this with the softmax. A representative example is that [83] proposed to learn a

face classify first, and then use the triplet constraint to fine-tune and boost the performance. It achieved promising accuracy in face recognition. Although we also integrate triplet information with the traditional classification objective, our method jointly optimizes these two objectives simultaneously, which is different from [83]. As shown in the experiments, this joint optimization strategy generates better feature representations for fine-grained image understanding. In addition, our algorithm can also easily support eleminating recognition irrelevant factors or embedding of label structures, e.g., hierarchy or shared attributes, which have been proven useful in various studies [6, 23, 2, 103, 118, 128, 12], but not well explored in learning fine-grained feature representation that can model similarity at different levels. Our proposed algorithm is elaborated in Section 3 and 4.

2.2 Hashing for Large-Scale Image Retrieval

Given fine-grained features extracted from massive image databases, either through the hand-crafted feature design or the above-mentioned deep learning-based feature engineering, the next goal is to index them to enable large-scale analysis (e.g., image retrieval) in real time. Recently, hashing methods have been intensively investigated in the machine learning and computer vision community for large-scale image retrieval [110]. Representative methods include, but are not limited to, locality-sensitive hashing [16] and its extension in kernel space [55, 56], spectral hashing [114], iterative quantization method [37], weakly-supervised hashing in kernel space [72], semi-supervised hashing [109] supervised hashing [66], compact kernel hashing with multiple features [67], and supervised discrete hashing [94].

Among these methods, kernelized and supervised hashing (KSH) [66] is generally considered the most effective, achieving state-of-the-art performance with a moderate training cost. Therefore, this was chosen in our framework for scalable image retrieval. The central idea of KSH is to reduce the gap between low-level hash code similarity and high-level semantic (label) similarity by virtue of supervised training. In doing so,

a similarity search in the binary code space can reveal the given semantics of examples. In other words, KSH does well in incorporating the given semantics into the learned hash functions or codes, while the other hashing methods are inadequate in leveraging the semantics. Specifically, compared to the unsupervised kernel hashing method [56, 66] and the semi-supervised linear hashing method [108, 109], KSH shows much higher search accuracy, as it takes full advantage of supervised information (originating from the semantics) that is not well exploited by those unsupervised and semi-supervised methods. Even compared against competing supervised hashing methods such as binary reconstructive embedding (BRE) [54] and minimal loss hashing (MLH) [79], KSH still shows clear accuracy gains yet with much shorter training time.

However, hashing methods, including KSH, tend to generate an unordered set for the same hash value, adversely affecting the classification accuracy when using majority voting, i.e., deciding the category of the query image via the retrieved images. This is particularly true for fine-grained image categorization, since the differences of these images are very subtle. Therefore, we propose the weighted hashing to alleviate this problem and it can accurately classify a large number of images, which will be elaborated in Section 6.

2.3 Content-Based Medical Image Retrieval

Medical images are special cases of fine-grained or even ultra-fine-grained images, since their differences (e.g., cancer grading) could be quite subtle and hard to differentiate even for human experts. Therefore, medical image analysis is chosen as the main use case to validate our algorithms. In fact, CBIR already shows its importance in medical image analysis by providing doctors with diagnostic aid in the form of visualizing existing and relevant cases, along with diagnosis information. Clinical decision-support techniques such as case-based reasoning or evidence-based medicine have a strong need for retrieving images that can be valuable for diagnosis.

For example, Comaniciu et al. [14] proposed a content-based image-retrieval sys-

tem that supports decision making in clinical pathology, in which a central module and fast color segmenter are used to extract features such as shape, area, and texture of the nucleus. System performance was assessed through a ten-fold cross-validated classification and compared with that of a human expert on a database containing 261 digitized specimens. Dy et al. [25] described a new hierarchical approach of CBIR based on multiple feature sets and a two-step approach. The query image is classified into different classes with best discriminative features between the classes. Then similar images are searched in the predicted class with the features customized to distinguish subclasses. El-Naqa et al. [26] proposed a hierarchical learning approach that consists of a cascade of a binary classifier and a regression module to optimize retrieval effectiveness and efficiency. They applied this to retrieve digital mammograms and evaluated it on a database of 76 mammograms. Greenspan et al. [38] proposed a CBIR system that consists of a continuous and probabilistic image-representation scheme. It uses GMM and information-theoretic image matching via the Kullback-Leibler (KL) measure to match and categorize X-ray images by body region. Song et al. [98] designed a hierarchical spatial matching–based image-retrieval method using spatial pyramid matching to effectively extract and represent the spatial context of pathological tissues. In the context of histopathological images from breast tissues, Schnorrenberg et al. [89] extended the biopsy analysis support system to include indexing and content-based retrieval of biopsy slide images. A database containing 57 breast-cancer cases was used for evaluation. Zheng et al. [144] designed a CBIR system to retrieve images and their associated annotations from a networked microscopic pathology image database based on four types of image features. Akakin et al. [1] proposed a CBIR system using the multi-tiered approach to classify and retrieve microscopic images, which enables both multi-image query and slide-level image retrieval in order to protect the semantic consistency among the retrieved images.

As emphasized in [145], scalability is the key factor in CBIR for medical image anal-

ysis. However, owing to the difficulties in developing scalable CBIR systems for large-scale data sets, most previous systems have been tested on a relatively small number of cases. With the goal of comparing CBIR methods on a larger scale, ImageCLEF and VISCERAL provide benchmarks for medical image-retrieval tasks [73, 74, 123, 59, 41]. Recently, Foran et al. [30] designed a CBIR system named ImageMiner for comparative analysis of tissue microarrays by harnessing the benefits of high-performance computing and grid technology. However, few attempts have been made to design computational and scalable retrieval algorithms in this area. Therefore, we introduce hashing methods and binary coding methods for large-scale medical image retrieval, elaborated in Section 5 and 6.

Anohter important respect in CBIR is integrating multiple features for accurate image retrieval. For example, accurate analysis of histopathological images requires to examine cell-level information for accurate diagnosis, including individual cells (e.g., appearance [9, 137] and shapes [24]) and architecture of tissue (e.g., topology and layout of all cells [3]). These features cover both local and holistic information, all benefiting the diagnosis accuracy of histopathological images. Therefore, the complementary descriptive capability of local and holistic features motivates us to integrate their strengths to yield more satisfactory results. The proposed graph-based query-specific fusion approach is described in Section 7.

CHAPTER 3: LEARNING FINE-GRAINED FEATURE REPRESENTATION WITH LABEL STRUCTURES

3.1 Motivation

Owing to the success of convolutional neural networks (CNN) [45, 53, 100, 95, 91], models of fine-grained image categorization have made tremendous progress in recognizing subtle differences among subordinate classes, such as different models of cars [50, 52, 65, 124], breeds of animals [49, 82, 18, 5, 51, 63, 107, 121], and types of food dishes [7, 126]. Most of previous methods focus on improving the classification accuracy, by learning critical parts that can align the objects and discriminate between neighboring classes [125, 10, 4, 130, 129, 34], or using distance metric learning to alleviate the issue of large intra-class variation [113, 105, 111, 85, 62].

However, such studies have rarely been dedicated to learn a structured feature representation that can discover similar images at different levels of relevance. Fig. 3.1 shows examples of similar cars from a fine-grained dataset [50]. Having the same fine-grained labels indicates exactly the same make, model and year, while cars are still similar even they have different labels, e.g., the same make but different year, or the same body style (e.g., SUV, Coupe) from different make. In other words, different fine-grained categories may still share the same semantic information, such as coarse-level labels or attributes. Such shared information in the hierarchy of similarity should be explored in fine-grained feature representation, since it is applicable to various use cases such as the recommendation of relevant products in e-commerce, e.g., products have to be visually and semantically similar, but not necessarily belong to the same fine-grained category.

To obtain the fine-grained feature representation, one solution is to incorporate

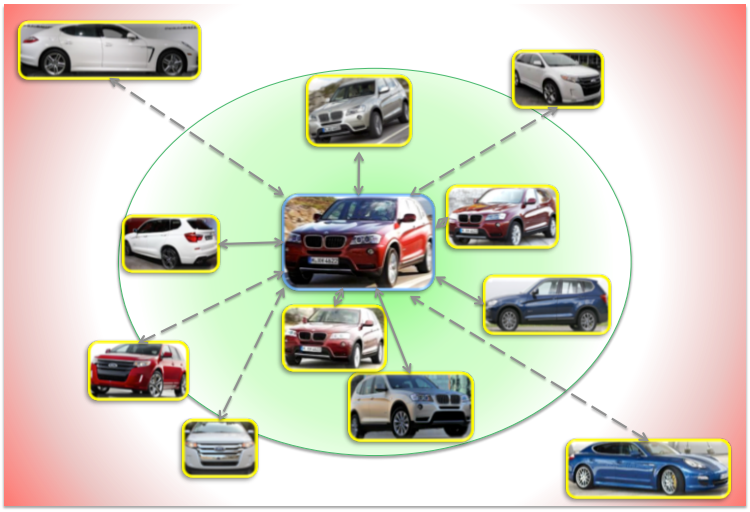


Figure 3.1: Examples from a fine-grained car dataset [50], where the similarity can be defined at different levels, i.e., body type, model, and even viewpoint, indicated by the distance to the query in the center. Images within the circle have exactly the same fine-grained labels, i.e., make and model, and the closest two also have the same viewpoint. Since images from different fine-grained categories may share the same coarse-level labels, such shared information should be leveraged to learn structured features.

similarity constraints (e.g., contrastive information [13] or triplets [80, 11]). For example, Wang et al. [111] proposed a deep ranking model to directly learn the similarity metric by sampling triplets from images. However, these strategies still have several limitations in fine-grained datasets: 1) Although the features learned from triplet constraints are effective at discovering similar instances, its classification accuracy may be inferior to the fine-tuned deep models that emphasize on the classification loss, as demonstrated in our experiments. In addition, the convergence speed using such constraints is usually slow. 2) More importantly, previous methods for fine-grained features do not leverage shared information in label structures, which is critical to locate images with relevance at different levels.

We propose two contributions to solve these issues: 1) A multi-task deep learning framework is designed to effectively learn the fine-grained feature representation without sacrificing the classification accuracy. Specifically, we *jointly optimize* the

classification loss (i.e., softmax) and the similarity loss (i.e., triplet) in CNN, which can generate both categorization results and discriminative feature representations. The integration of two constraints not only boosts the classification accuracy, but also produces effective features that are able to discover visually and semantically similar instances in fine-grained datasets. 2) Furthermore, based on this framework, we propose to seamlessly *embed label structures* such as hierarchy (e.g., make, model and year of cars) or attributes (e.g., ingredients of food), which is achieved by designing generalized triplets. Therefore, shared information in label structures (e.g., same coarse-level labels or same attributes) can be effectively leveraged as extra constraints and augmented data. Such strategy of embedding label structures is able to effectively discover relevant images with respect to different levels of similarities. We evaluate our methods on four fine-grained datasets, i.e., the Stanford car, the Car-333, the CUB200-2011 and a fine-grained food dataset, containing either hierarchical labels or shared attributes. The experimental results demonstrate that our feature representation can precisely differentiate fine-grained or subordinate classes, and also effectively discover similar images at different levels of relevance, both of which are challenging problems.

3.2 Methodology

In this section, we introduce the joint optimization strategy for learning fine-grained feature representation. Then, we extend the algorithms to effectively embed structured labels, such as the hierarchy or shared attributes. The overall framework is shown in Fig. 3.2. We also provide important details in terms of implementation.

3.2.1 Multi-Task Learning for Joint Optimization

Traditional classification constraints such as softmax with loss are usually employed in CNN for fine-grained image categorization, which can distinguish different subordinate classes with high accuracy. Suppose that we are given N training images

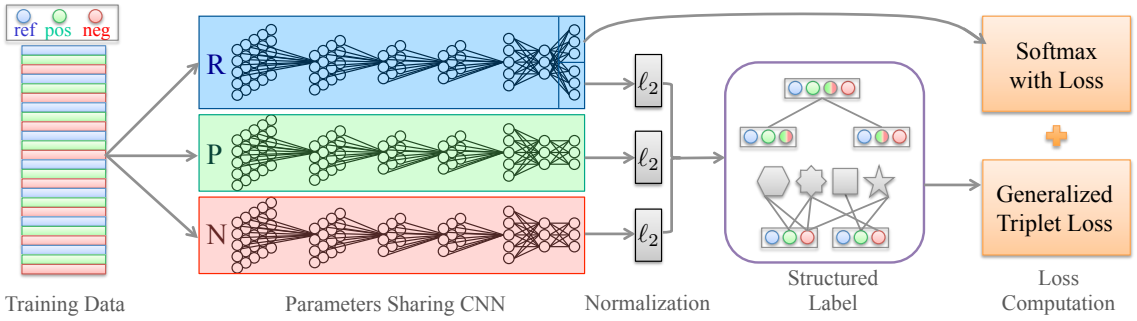


Figure 3.2: Our framework takes the triplets (i.e., the reference, the positive and the negative images) and the label of the reference image as the input, which pass through the three networks with shared parameters. The label structures are embedded in the loss layer, including the hierarchy or shared attributes. Two types of losses are optimized jointly to obtain the fine-grained classifier and also the feature representation.

$\{r_i, l_i\}_{i=1}^N$ of C classes, where each image r_i is labeled as class l_i. Given the output of the last fully connected layer $f_s(r_i, c)$ for each class $c = 1, \cdots, C$, the loss of softmax can be defined as the sum of the negative log-likelihood over all training images $\{r_i\}_i$:

$$E_s(r, l) = \frac{1}{N} \sum_{i=1}^{N} -\log \underbrace{\frac{e^{f_s(r_i, l_i)}}{\sum_{c=1}^{C} e^{f_s(r_i, c)}}}_{P(l_i|r_i)}, \tag{3.1}$$

where $P(l_i|r_i)$ encodes the posterior probability of the image r_i being classified as the l_ith class. In a nutshell, Eq. 3.1 aims to "squeeze" the data from the class into a corner of the feature space. Therefore, the intra-class variance is not preserved, while

Algorithm 1: Joint Optimization Framework.

Input : Training samples r_i, p_i, n_i, learning rate $\eta(t)$

while *not converge* **do**
- $t \leftarrow t + 1$;
- Calculate $f_s(r_i), f_t(r_i), f_t(p_i), f_t(n_i)$ by forward propagation ;
- Calculate $\nabla W_s = \lambda_s \cdot \frac{\partial E_s(r_i, l_i)}{\partial W_s}$;
- Calculate $\nabla W_t = (1 - \lambda_s) \cdot \frac{\partial E_t(r_i, p_i, n_i, m)}{\partial W_t}$;
- $\nabla W = \nabla W_s + \nabla W_t$;
- Update $W = W + \eta(t) \nabla W$

end

Output: Parameters W

such variance is essential to discover both visually and semantically similar instances.

To address these limitations, we explicitly model the similarity constraint in CNN using a multi-task learning strategy. Specifically, the triplet loss is fused with the classification objective as the similarity constraint. A triplet consists of three images, denoted as (r_i, p_i, n_i), where r_i is the reference image from a specific class, p_i an image from the same class, and n_i an image from a different class[1]. Given an input image r_i (similarly for p_i and n_i), this triplet-driven network can generate a feature vector $f_t(r_i) \in \mathbb{R}^D$, where the hyper-parameter D is the feature dimension after embedding. Ideally, for each reference r_i, we expect its distance from any n_i of different class is larger than p_i within the same class by a certain margin $m > 0$, i.e.,

$$\mathscr{D}(r_i, p_i) + m < \mathscr{D}(r_i, n_i), \tag{3.2}$$

where $\mathscr{D}(\cdot, \cdot)$ is the squared Euclidean distance between two ℓ_2-normalized vectors $f_t(\cdot)$ of the triplet network. To enforce this constraint in CNN training, a common relaxation [80] of Eq. 3.2 can be defined as the following hinge loss:

$$\begin{aligned} &E_t(r, p, n, m) = \\ &\frac{1}{2N} \sum_{i=1}^{N} \max\{0, \mathscr{D}(r_i, p_i) - \mathscr{D}(r_i, n_i) + m\}. \end{aligned} \tag{3.3}$$

In the feature space defined by $f_t(\cdot)$, it can group the r and p together while repelling the n by minimizing $E_t(r, p, n, m)$. The gradient can be computed as:

$$\begin{aligned} \nabla W_t = &2(f_t(r_i) - f_t(p_i)) \frac{\partial f_t(r_i) - \partial f_t(p_i)}{\partial W_t} \\ &- 2(f_t(r_i) - f_t(n_i)) \frac{\partial f_t(r_i) - \partial f_t(n_i)}{\partial W_t}, \end{aligned} \tag{3.4}$$

[1]Note that such triplet loss has been used in feature embedding [90, 111], but not with a joint optimization strategy or label structures.

if $\mathscr{D}(r_i, n_i) - \mathscr{D}(r_i, p_i) < m$, otherwise 0. Different from the pairwise contrastive loss [13] that forces the data of the same class to stay close with a fixed margin, the triplet loss allows certain degrees of intra-class variance. Despite its merits in learning feature representation, minimizing Eq. 3.3 for recognition tasks still has several disadvantages. For example, given a dataset with N image, the number of all possible triplets is N^3, and each triplet contains much less information (i.e., similar or dissimilar constraints with margins) compared with the classification constraint that provides a specific label among C classes. This can lead to slow convergence. Furthermore, without the explicit constraints for classification, the accuracy of differentiating classes can be inferior to the traditional CNN using softmax, especially in fine-grained problems where the differences of subordinate classes are very subtle.

Given the limitations of training with the triplet loss (Eq. 3.3) solely, we propose to jointly optimize two types of losses using a multi-task learning strategy. Fig. 3.2 shows the CNN architecture of our joint learning. The R, P, N networks share the same parameters during training. After the ℓ_2 normalization, the outputs of the three networks (i.e., $f_t(r), f_t(p), f_t(n)$) are transmitted to the triplet loss layer to compute the similarity loss $E_t(r, p, n, m)$. In the meantime, the output of the network R, $f_s(r)$, is forwarded to the softmax loss layer to compute the classification error $E_s(r, l)$. Then, we integrate these two types of losses through a weighted combination:

$$E = \lambda_s E_s(r, l) + (1 - \lambda_s) E_t(r, p, n, m), \tag{3.5}$$

where λ_s is the weight to control the trade-off between two types of losses. This framework of unifying three networks through Eq. 3.5 not only learns the discriminative features but also preserves the intra-class variance, without sacrificing the classification accuracy. In addition, it resolves the issue of the slow convergence when only using the triplet loss. We optimize Eq. 3.5 using the standard stochastic gradient

descent with momentum. The optimization procedure is summarized in Algorithm 1. Regarding the sampling strategy, one can either follow the methods in FaceNet [90], or employ hard mining approaches to explore challenging examples in the training data. Both of them are effective in our framework, since jointly optimizing $E_s(r, l)$ facilitates the searching of good solutions, allowing certain flexibility for the sampling.

During the testing stage, this framework takes one image as an input, and generates the classification result through the softmax layer, or the fine-grained feature representation after the ℓ_2 normalization. This discriminative feature representation can be employed for various tasks such as classification, verification and retrieval, which is more effective than solely optimizing the softmax with loss.

3.2.2 Embed Label Structures

As discussed before, an effective feature representation should be able to search relevant instances at different levels (e.g., Fig. 3.1), even not within the same fine-grained class. Our multi-task framework serves as a baseline to naturally embed label structures, without sacrificing the classification accuracy on fine-grained datasets. In particular, we aim to handle two types of label structures, i.e., hierarchical labels and shared attributes, both of which have wide applications in practice.

3.2.2.1 Generalized Triplets for Hierarchical Labels

In the first case, the fine-grained labels can be naturally grouped in a tree-like hierarchy based on semantics or domain knowledge. The hierarchy can contain multiple levels. For simplicity purpose, we explain the algorithm with a two-level structure, and then generalize to multiple levels. Fig. 3.3 illustrates an example of two-level labels from a car dataset [50], where the fine-grained car models in the leaf nodes are grouped according to their body types in the roots. Two cars with different fine-grained categories may share the same coarse-level label, e.g., both of them are SUV or Sedan in Fig. 3.3. Intuitively, sharing the same body types should attain higher

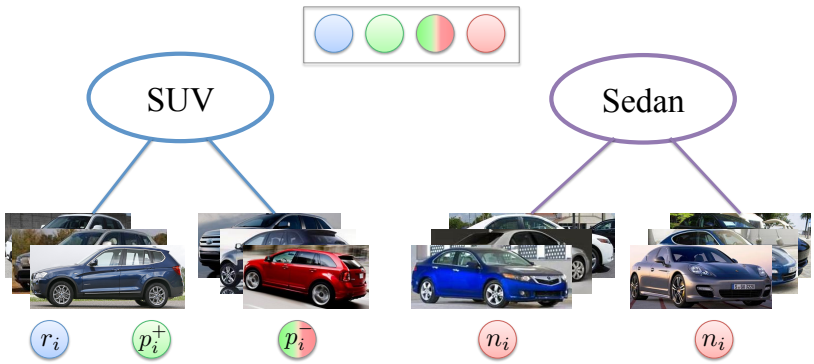


Figure 3.3: The hierarchy of labels in the fine-grained car dataset [50]. Blue (r_i) means the reference image, green (p_i^+) denotes the image with the same fine-grained label (i.e., the same make, model and year), green-red (p_i^-) represents different fine-grained labels but the same coarse label (i.e., the body type), and red (n_i) indicates different coarse labels.

similarity than having different ones.

To model such shared information in the hierarchy of coarse and fine class labels, we propose to generalize the concept of triplet. Specifically, *quadruplet* is introduced to model the two-level structure. Each quadruplet, (r_i, p_i^+, p_i^-, n_i), consists of four images. Similar to triplet, p_i^+ denotes the image of the same fine-grained class as the reference r_i. The main difference is that in quadruplet, all negative samples are classified into two sub-categories: the more similar one p_i^- that shares the same coarse class with r_i, and the more different one n_i sampled from different coarse classes. Given a quadruplet, this hierarchical relation among the four images can be described in two inequalities,

$$\mathscr{D}(r_i, p_i^+) + m_1 < \mathscr{D}(r_i, p_i^-) + m_2 < \mathscr{D}(r_i, n_i), \tag{3.6}$$

where the two hyper-parameters, m_1 and m_2, satisfying $m_1 > m_2 > 0$, control the distance margins across the two levels. It is worth to mention that if Eq. 3.6 is satisfied, then $\mathscr{D}(r_i, p_i^+) + m_1 < \mathscr{D}(r_i, n_i)$ automatically holds. Compared to triplet, quadruplet is able to model much richer label structures between different levels, i.e., coarse labels and fine-grained labels. As a result, the learned feature representation

can discover relevant instances that are appropriate in specific scenarios, e.g., locating a car with specific model and year, or finding SUVs from different body types.

Regarding the sampling strategy, all training images are used as the references in every epoch. For each reference image r_i, we select p_i^+, p_i^- and n_i from other corresponding classes, depending on both fine and coarse labels. To incorporate this quadruplet constraint in CNN training, we propose to decompose Eq. 3.6 into two triplets, (r_i, p_i^+, p_i^-) and (r_i, p_i^-, n_i), phrased as *generalized triplets*. Similar to Eq. 3.3, our approach seeks for the optimal parameters that minimize the joint loss over the sampled quadruplets:

$$\begin{aligned} &E_q(r, p^+, p^-, n, m_1, m_2) = \\ &\frac{1}{2N}\sum_{i=1}^{N}\max\{0, \mathscr{D}(r_i, p_i^+) - \mathscr{D}(r_i, p_i^-) + m_1 - m_2\} \\ &\quad + \frac{1}{2N}\sum_{i=1}^{N}\max\{0, \mathscr{D}(r_i, p_i^-) - \mathscr{D}(r_i, n_i) + m_2\}. \end{aligned} \tag{3.7}$$

Clearly, this generalized triplets can be naturally incorporated into our multi-task learning framework (Eq. 3.5).

So far we have mainly discussed in the scenario of a two-level label hierarchy, through the generalized triplet representation of quadruplet. In fact, our method is also applicable to the more general multi-level case using the same strategy, i.e., representing a "tuplet" with generalized triplets. Similar to the quadruplet sampling strategy, each tuplet is formed by selecting the classes at different similarity levels, from which training images are sampled (one image at each level). Therefore, a tuplet from an x-level hierarchy contains $x + 2$ images (e.g., the quadruplet from a two-level hierarchy has four images). This tuplet is decomposed into x triplets, by taking the reference image and two more images from two adjacent levels. Intuitively, this means that multiple triplets are sampled to represent different levels of similar-

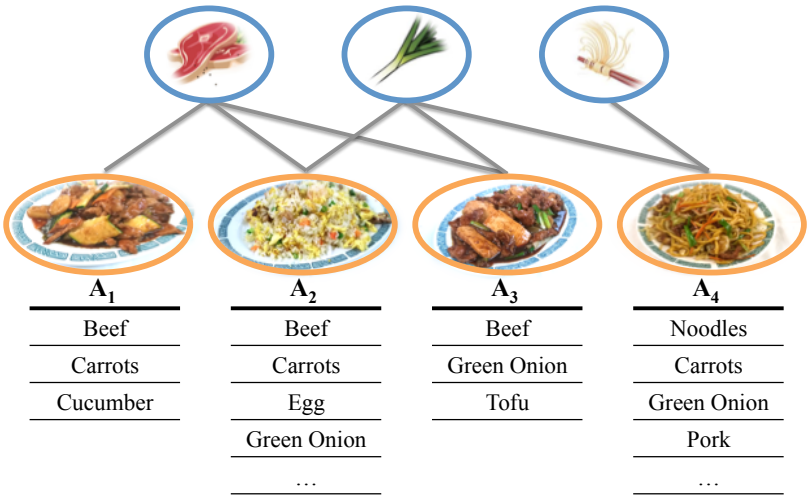


Figure 3.4: The shared attributes in our food dataset, where the attributes (A_1-A_4) mean the ingredients.

ity, i.e., images with the same finer-level labels are more similar than ones sharing the same coarser-level labels. Same as the two-level case, it can be optimized using the multi-task learning framework based on triplets. Even though this is not exhaustive sampling or exact decomposition for the tuplet, the generalized triplets are representative enough to ensure a good performance, which is demonstrated in our experiments (Section 3.3.2). It is also worth mentioning that the traditional triplet is a special case of the generalized triplet, i.e., only one-level hierarchy.

3.2.2.2 Generalized Triplets for Shared Attributes

In the second case, fine-grained objects can share common attributes with each other. For instance, Fig. 3.4 illustrates that fine-grained food dishes can share some ingredients, indicating relevance at different levels. Intuitively, classes that share more attributes should be more similar than the classes sharing less attributes. Therefore, such shared information should also be explored in our multi-task learning framework. Unlike the tree-like hierarchy in the first case, we are not able to directly model the label dependency as Eq. 3.6, because some fine-grained classes can own multiple attribute labels. Instead, we model this graph dependency using a modified triplet idea. To have a better understanding of our method, we can consider the first three

dishes shown in Fig. 3.4. Although both the second and third dishes belong to different classes compared to the first one, the second dish shares more attributes (beef, carrots) with the first dish. This difference in attribute overlapping inspires us to re-define the margin m, i.e., the distance between $\mathscr{D}(r_i, p_i)$ and $\mathscr{D}(r_i, n_i)$, as the Jaccard similarity [43] of attributes from different classes:

$$m = m_b \left(1 - \frac{|A_p \cap A_n|}{|A_p \cup A_n|}\right), \tag{3.8}$$

where m_b is a constant factor specified as the base margin, A_p and A_n are the sets of attributes belonging to the positive and negative categories, respectively. Therefore, the more attributes these classes share, the smaller margin this triplet has. Using such adaptive margin for the triplet loss, the learned feature can discover images containing common attributes as the query images. Similarly, Eq. 3.8 can be naturally incorporated in our multi-task learning framework based on the triplet loss. In fact, the original triplet constraint is also a special case of the multi-attribute constraint, when each fine-grained label only connects to one attribute, i.e., no shared labels.

3.2.3 Implementation Details

In terms of implementation details, all CNNs are based on GoogLeNet [100], and are fine-tuned on these fine-grained datasets for the best performance and fair comparisons. Note that our method is very general and also applicable to other networks, such as AlexNet [53] or VGGNet [95], discussed in the experiment section.

The input data is organized in the following way. A list, (r_i, p_i, n_i, l_i, m), is generated and can be dynamically updated during runtime, in which l_i is the label of image r_i, and m is the margin that is fixed in traditional triplet loss or adjustable as per the hierarchical or shared attribute structure in our proposed method. Such data is sent into R, P, N networks (Fig. 3.2), which share the parameters during the training procedure. Different from P and N, the last fully connected layer of R is connected

with two modules. One is combined with the label l_i to compute the softmax with loss, and the other is sent to the ℓ_2 normalization layer to generate $f_t(r_i)$ for the generalized triplet loss. Finally, the two losses are combined by the weight λ_s and used for back propagation.

Regarding our hyper-parameters, we empirically set the feature dimension as 200, the margin m and base margin m_b as 0.2, and the weight λ_s as 0.8, with discussions of the parameter tuning and sensitivity in the experiment section.

3.3 Experiments

In this section, we conduct thorough experiments to evaluate this proposed framework on four fine-grained datasets with label structures. We aim to demonstrate that our learned feature representations can discover similar instances at different levels of relevance, without sacrificing the accuracy of differentiating fine-grained or subordinate classes. To this end, CNNs are chosen as the baseline, owing to its tremendous success in fine-grained image categorization. Note that we do not emphasize on the comparison with part-based systems [130, 34, 125, 129], since our method does not use any information from parts, i.e., the scope of the research is different. Particularly, we have evaluated the retrieval precision of four methods that can generate fine-grained feature representation: 1) deep feature learning by triplet loss [90, 111], 2) triplet-based fine-tuning after softmax [83], i.e., not joint optimization, 3) our multi-task learning framework, and 4) our framework with label structures. In terms of the classification task, besides these four methods, we also report the accuracy of using CNN with traditional softmax. We carefully follow the specifications from these compared papers for their settings and parameters.

3.3.1 Stanford Car with Two-Level Hierarchy

The first experiment focuses on the efficacy of embedding hierarchical labels, using the Stanford car dataset [50]. It contains 16,185 images (with bounding boxes) of 196

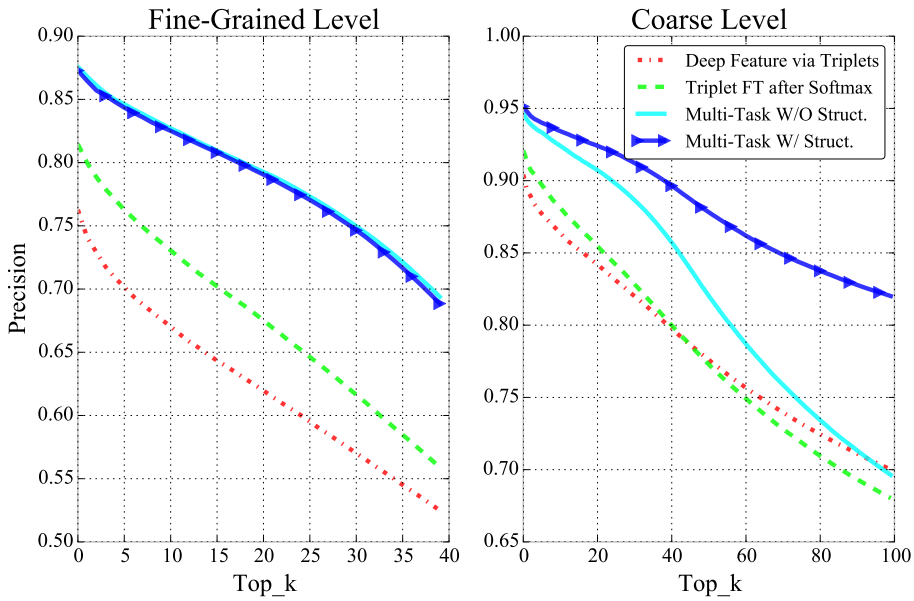


Figure 3.5: Comparison of retrieval precision on the Stanford car, with two levels of labels.

car categories, with 8,144 for training and the rest for testing. The categories, i.e., fine-grained class labels, are defined as make, model and year, such as Audi S4 Sedan 2012. Following [50], we have assigned each fine-grained label to one of nine coarse body types, such as SUV, Coupe and Sedan (Fig. 3.3 in [50]), resulting in a two-level hierarchy.

Fig. 3.5 shows the retrieval precision using feature representations extracted by various CNNs, at both the fine-grained level and the coarse level. At the fine-grained level, results from our multi-task learning methods are better than the others, i.e., at least 13.5% higher precision at top-40 retrievals (using top-40 since each fine-category has around 40 images). The reason is that the joint optimization strategy leverages the similarity constraints via triplets, which can augment the training information, assisting the network to reach better solutions. No matter using the traditional or generalized triplets (i.e., without or with the label structures) in our framework, the difference of precision is within 0.5%, which can be caused by the sampling strategies. At the coarse level, our method without label structures also fails to achieve high precision at top-100 retrievals, while using generalized triplets significantly outperforms

(a) Without Label Structures (b) With Label Structures

Figure 3.6: Visualization of features after dimension reduction. Different colors represents different coarse-level labels, and intensities (or transparency) from the same color indicate fine-grained labels.

Table 3.1: Comparison of the classification accuracy on four fine-grained datasets, from methods following the similar framework as ours. The best result in each column is highlighted. Note that *embedding label structures aims to enhance the retrieval precision (our main contribution, shown in Fig. 3.5, 3.7, 3.8 and 3.9)*, while the improvement of classification may depend on datasets. Overall our classification results of joint optimization with or without label structures are 1.5-10% higher than works under the similar framework.

	Stanford Car	**Car-333**	**CUB200-2011**	**Food**
Softmax with Loss	86.9%	87.9%	76.7%	87.1%
Traditional Triplets	78.7%	61.2%	72.4%	78.2%
Triplet FT after Softmax	83.0%	81.7%	75.3%	86.1%
Multi-Task W/O Struct.	**88.4%**	88.9%	78.2%	88.5%
Multi-Task W/ Struct.	88.3%	**89.4%**	**78.8%**	**89.0%**

the others, i.e., at least 12.4% higher precision, demonstrating the efficacy of our embedding scheme. To provide insights of our promising results on this coarse-level retrieval, we extract features from our multi-task learning framework using traditional and generalized triplets, and visualize them in Fig. 3.6 after dimension reduction. Six coarse-level classes are randomly chosen, and five fine-level classes are sampled from each coarse one. The features from generalized triplets are consistently much better separated than ones from traditional triplets, benefited from the embedding of label structures.

Table 3.1 shows the classification accuracy of CNN methods using the Stanford Car

dataset, comparing with methods following similar framework as ours. A fine-tuned GoogleNet achieves 86.9%. Learning deep features via triplets alone [90, 111] attains 78.7%, which is worse than GoogleNet. The reason is that softmax with loss can explicitly minimize the classification error, while triplets attempt to implicitly separate classes by constraining the similarity measures. Fine-tuning with triplets after the softmax [83] also aims to integrate the classification and similarity constraints, same as ours. This identification and verification framework achieves promising performance in face recognition. However, different from our framework, it embeds the triplet loss after learning a face classifier, i.e., not a joint optimization strategy as ours. This may adversely affect the classification accuracy in fine-grained image categorization, since triplet loss only implicitly constrains the classification error, which may not be sufficient in further differentiating subordinate classes during fine-tuning. As a result, it achieves 83.0%, which is worse than the fine-tuned GoogleNet. Our multi-task learning framework achieves 88.3% and 88.4% when jointly optimizing both types of losses[2], which are higher than the other methods for learning feature representations, and among state-of-the-art that do not use parts. Note that our methods with or without the label structures have very similar accuracy for the fine-grained classes, since the purpose of embedding label structures is to discover similar instances at different levels of relevance, i.e., our main contribution shown in (Fig. 3.5), not to improve the fine-grained classification. It is possible that such augmented information can benefit the classification process (demonstrated on the other datasets), while this is not always guaranteed.

Note that part-based models still achieve the best classification accuracy (e.g., 92.8% in [51] using around thirty parts), owing to the discriminative part regions and augmented training data. However, these methods possibly require additional labels and/or computational time to train networks for parts, while our framework takes

[2]Classification in our framework is achieved by using the extracted features with Support Vector Machine (SVM) or k-nearest neighbors, both of which achieve very similar accuracy.

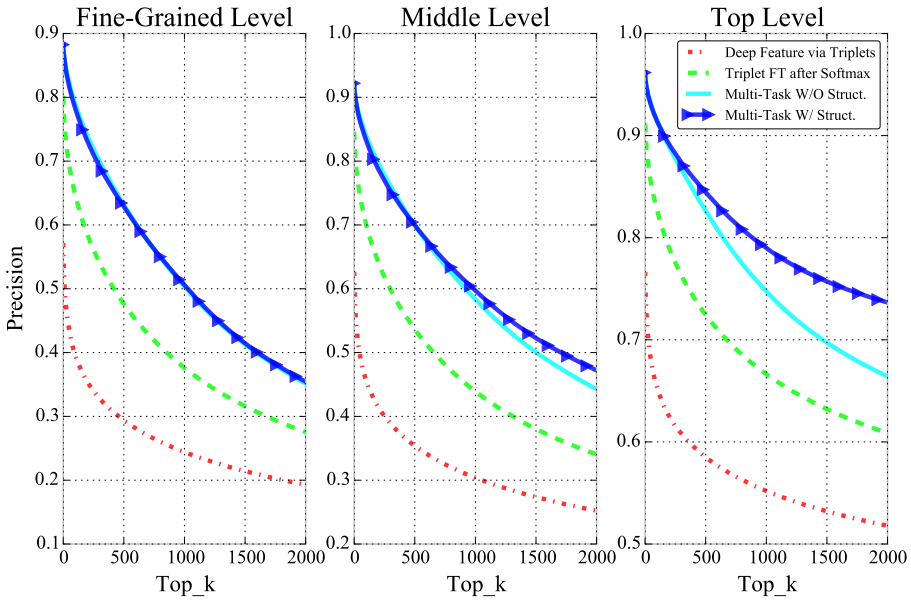


Figure 3.7: Comparison of retrieval precision on the Car-333 dataset. Top-level means the car make only. Mid-level represents both make and model. Fine-level denotes the fine-grained labels of make, model and year range.

whole images as input to learn feature representations. We believe that two directions of research, i.e., part-based methods and our framework, are both important, and can potentially benefit each other.

3.3.2 Car-333 with Three-Level Hierarchy

The second experiment also investigates the hierarchical labels, but using a much larger car dataset [118] to validate the scalability. These are end-user photos from the Craigslist, so they are more naturally photographed. It contains 157,023 training images and 7,840 testing images, from 333 car categories. The categories are defined by make, model and year range. Note that two cars of the same model but from different years are considered as different classes. The bounding boxes are generated by Regionlets [112], which produces promising results in car detection. Different from the Stanford car, this has a three-level hierarchy: 333 fine-grained labels are grouped into 140 models by ignoring the difference of years, and then five makes (i.e., Chevrolet, Ford, Honda, Nissan, Toyota).

Fig. 3.7 shows the retrieval precision at these three levels. Since the training data

is around 20 times larger than the previous one, we show the precision upon top-2000 retrievals (note that the number of images in a fine-level class can be less than 2000). The results are consistent with the ones on the Stanford car, demonstrating that the strategy of generalized triplets is applicable to multi-level hierarchies. Specifically, our method with label structures is at least 13.2% better than other methods in terms of the top-2000 retrieval precision at the middle level, and 12.8% better at the top level. This is also 7.2% better than ours without embedding structures at the top level, proving the efficacy of our generalized triplets. In addition, such promising results also demonstrate that the scalability of our methods such as generalized triplets is sound. Regarding the classification accuracy (summarized in Table 3.1), GoogleNet achieves 87.9%, the deep feature via traditional triplets attains 61.2%, fine-tuning with triplets after softmax reaches 81.7%. It is worth mentioning that the deep feature via triplets has considerably worse performance on this dataset, compared to the results on the Stanford car. It indicates that this method does not have good scalability for fine-grained image categorization, although it is proven to be effective for other tasks such as verification and ranking [90, 111]. On the other hand, jointly optimizing the softmax with loss can alleviate this issue even on this larger-scale dataset, as it directly tackles the classification problem. Using this strategy, our method achieves 89.4%, which is among state-of-the-art.

3.3.3 CUB200-2011 Dataset with Shared Attributes

The third experiment aims to examine the embedding of shared attributes, the CUB200-2011 [104], which contains attributes information. Particularly, this dataset has 5,994 training and 5,794 testing images, 200 classes and 312 attributes, with bounding boxes provided. Fig. 3.8 shows the retrieval precision on this bird dataset with respect to top-30 retrievals. In addition to evaluate on the fine-grained labels, we also define a new level of relevance: two images are similar when they share at least 50% of the attributes, since bird dataset has a large amount of attributes. Our method

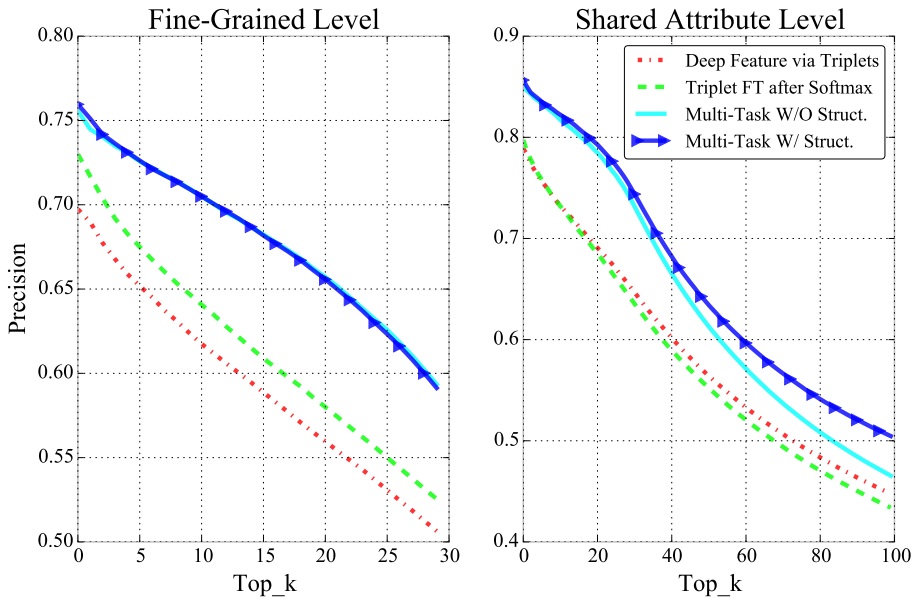


Figure 3.8: Comparison of retrieval precision on the CUB200-2011 dataset [104]. Share Attribute Level means that two images are relevant if they share at least 50% of the attributes, since bird dataset has a large amount of attributes.

by embedding shared attributes outperforms the others by 6% at the fine-grained level, and 4% at the attribute level in terms of the precision. This is consistent with the other datasets, i.e., label structures can significantly improve the retrieval precision at different levels of relevance. The classification results are listed in Table 3.1. Note that part-based methods can achieve above 85% accuracy [8], which is better than methods without using parts. Again, our method has different aims. It is mainly designed for learning the fine-grained feature representation, which considerably improves the image retrieval precision at different levels of label structures (Fig. 3.8), while still attaining promising classification accuracy among methods that do not rely on parts.

3.3.4 Food Dataset with Shared Attributes

The fourth experiment aims to examine our newly collected food dataset that consists of ultra-fine-grained classes and rich class relationships. To generate this dataset, we sent multiple data collectors to six restaurants, and they took photos of most dishes during two months. In total, we acquired 37,086 food photos from 975 menu items, i.e., fine-grained class labels. In addition, we built a list of 51 ingredients,

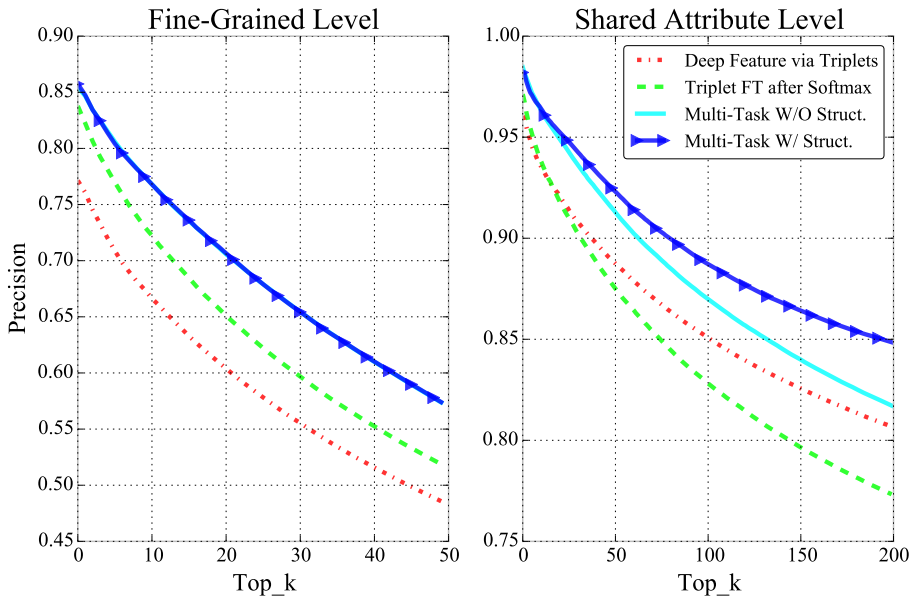


Figure 3.9: Comparison of retrieval precision on the food dataset. Share Attribute Level means that two images are relevant if they share at least one attribute.

i.e., shared attributes, to precisely describe these dishes. This dataset is divided into 32,135 training and 4,951 testing images, and testing images are collected on different days from the training, to mimic a realistic scenario by avoiding potential correlations of taking photos in the same day (e.g., multiple photos from the same dish at the same time cannot be used for both training and testing).

Fig. 3.9 shows the retrieval precision on this food dataset with respect to top-50 retrievals, as each category has around 20 to 50 images. In terms of the relevance based on attributes, we define that two images are similar when they share at least one attribute. Our method by embedding shared attributes outperforms the others by 5.5% at the fine-grained level, and 4.2% at the attribute level in terms of the precision. Since the precisions of these methods are already above 80%, such improvement means a reducing of 21.7% for the errors. Compared to our method without embedding attributes, it is nearly the same performance at the fine-grained level, while 3.1% better at the attribute level (reducing errors by 16.9%), demonstrating the efficacy of the generalized triplets with adaptive margins. Note that the improvement may not be as significant as on the other two datasets using hierarchical labels. The reason is

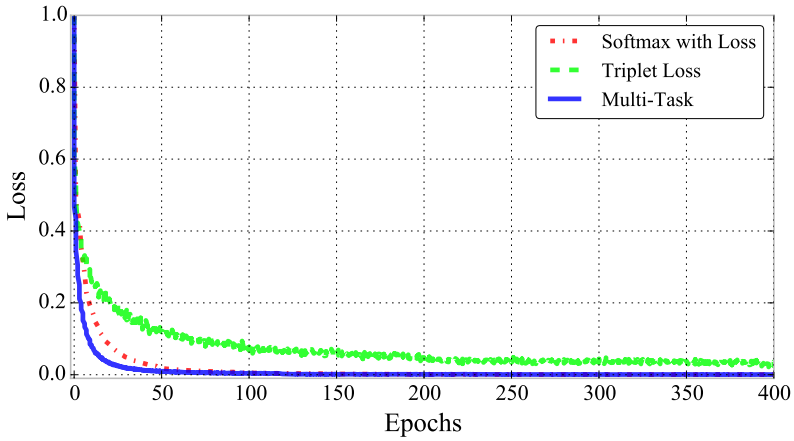


Figure 3.10: Comparison of the convergence rate on the Stanford car dataset. The first 400 epoches are shown for better visualization.

that the similarity measure for attributes is more subtle, i.e., two cars having different coarse labels could be more distinguishable than two dishes sharing no attributes. In terms of the classification accuracy (summarized in Table 3.1), we have achieved 89.0%, comparing to 87.1% by GoogleNet, 78.2% by learning the deep feature and 86.1% by fine-tuning with triplets after softmax. This is also a promising result, considering that this challenging dataset is ultra-fine-grained.

3.3.5 Discussions

In this section, we discuss the convergence rate, parameter sensitivity and applicability to other networks.

Convergence rate: Fig. 3.10 shows the convergence rate of these methods. Since each triplet contains much less information compared to the one of using the label directly (i.e., softmax with loss), their convergence rates can be dramatically different. Particularly, using softmax with loss has much faster convergence rate than using triplet loss. Our multi-task learning framework jointly minimizes both of them, so it harvests augmented information from both sides, resulting in a fast convergence rate as well. Overall, our methods converge after 800 epochs on the Stanford car, 150 epochs on the Car-333, and 600 epochs on the food dataset, which are reasonably fast in practice.

Parameter tuning: Our framework has one important parameter, the weight λ_s to balance two types of losses, and setting λ_s to be 0 or 1 degenerates our framework to deep feature learning by triplet loss [90, 111] or GoogleNet (softmax with loss), respectively, which will either fail to differentiate fine-grained classes or lose the ability to generate effective feature representations. Since softmax with loss may contain more information than a triplet in each iteration, it is reasonable to assign a higher weight to softmax, i.e., larger than 0.5. Our experiments show that the performance is not sensitive to small variations to λ_s, i.e., within 0.8% difference in a range of $[0.55, 0.85]$. Besides the weight, the feature dimension and the margin is also relevant to the classification accuracy, while they are less important compared to the λ_s. From our extensive experiments, we observe that our methods are also stable with respect to their variations up to a certain range, e.g., within 2% difference for feature dimensions from 128 to 512. Therefore, it is relatively easy to tune the hyper-parameters in our framework. In fact, we use the same group of parameters on all datasets.

Other networks: Although we build our network based on GoogleNet for most experiments, our proposed strategies are also applicable to other types of networks [53, 95]. For example, based on AlexNet [53], our framework with label structures achieves 79.6% classification accuracy and 82.2% precision at the coarse-level for top-100 retrievals, which are 3.0% and 9.9% higher than using the traditional AlexNet fine-tuned on this dataset, demonstrating the efficacy of our strategies on a different network architecture. Note that without fine-tuning, the DeCAF model [20] from AlexNet [53] and Imagenet [17] has much worse performance than the fine-tuned one, due to the difference between the generic Imagenet and the specific fine-grained datasets, which confirms the challenges of this problem.

Retrieved images Fig. 3.11, Fig. 3.12, Fig. 3.13 and Fig. 3.14 show the top 1-5 retrieved images, and also images after top 100. In these figures, top 1-5 are usually from the same fine-grained category, owing to their high classification accuracy. Im-

ages after top 100 usually have no ones from exactly the same fine-level class, which is not surprising since the number of images in each class is limited (e.g., around 40 in the Stanford car dataset). Without label structures, it is likely to retrieve visually similar but semantically irrelevant images. Our method with label structures successfully discovers relevant images at the coarse-level, e.g., the same body type in the Stanford car dataset, or the same make in the car-333 dataset, proving that our fine-grained feature representation is able to accurately differentiate subordinate classes, and also effectively search similar images with respect to different levels of relevance. We believe that our framework has various use cases, including the recommendation of relevant products in e-commerce. For example, consumers may be interested in food with similar ingredients, even if they are not the same dish (Fig. 3.14).

3.4 Summary

In this chapter, we introduce a multi-task learning framework to effectively generate fine-grained feature representations by embedding label structures, such as hierarchical labels or shared attributes. In our method, the label structures are seamlessly embedded in CNN through the proposed generalized triplets, which can incorporate the similarity constraints at different levels of relevance. Such a framework retains the classification accuracy for subordinate classes with subtle differences, and at the same time considerably improves the image retrieval precision at different levels of label structures on three fine-grained datasets, including a newly-collected benchmark dataset for food. These merits warrant further investigating the embedding of label structures for learning fine-grained feature representation.

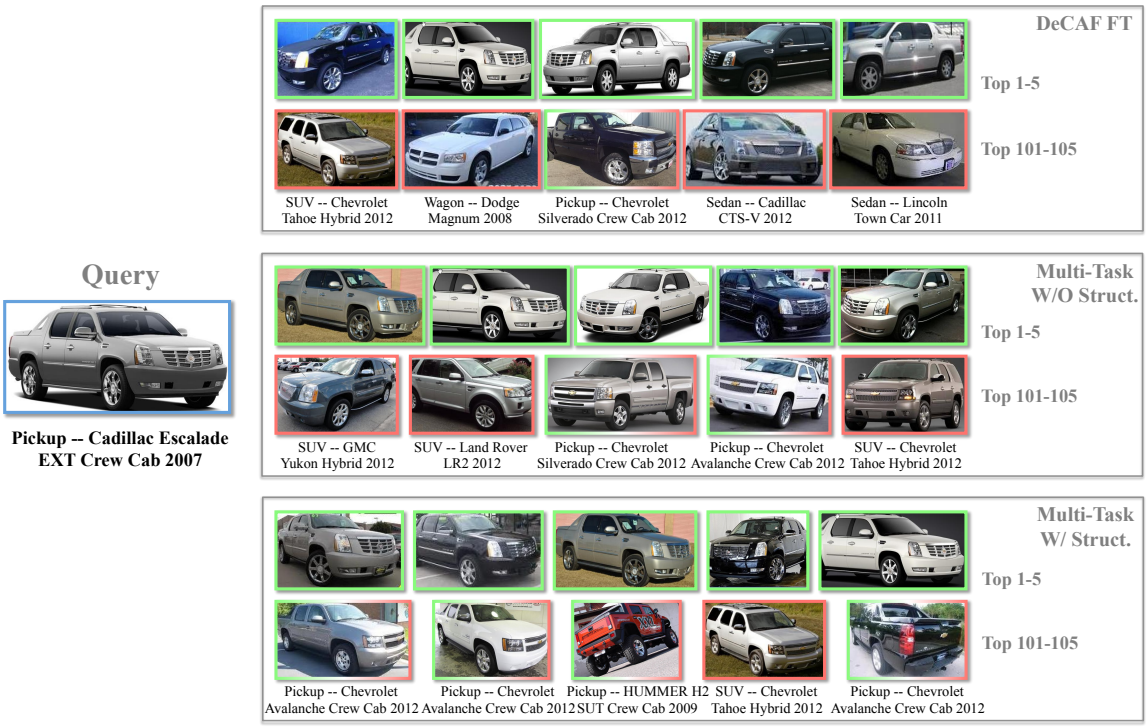


Figure 3.11: Retrieved images in the Stanford car dataset. Green means the same fine-grained category, green-red means different fine-grained but the same coarse category (the ratio between green and red indicates similarity scores), and red means different coarse category. DeCAF [20] FT means that we fine tune the AlexNet [53] on this dataset, and then extract features from its fc_7 layer, i.e., feature representation from softmax with loss. In other words, it only relies on the classification constraint for training. Therefore, its retrieved images are not visually similar to the query, even though they have the same fine-grained label. Contrarily, images retrieved by our methods, which jointly optimize the triplet loss, are more visually similar.

Figure 3.12: Retrieved images in the Car-333 dataset. Green means the same fine-grained category, green-red means different fine-grained but the same coarse category (the ratio between green and red indicates similarity scores), and red means different coarse category.

Figure 3.13: Retrieved images in the CUB200-2011 dataset. Green means the same fine-grained category, green-red means different fine-grained but the same coarse category (the ratio between green and red indicates similarity scores, i.e., Jaccard similarity), and red means sharing no attributes.

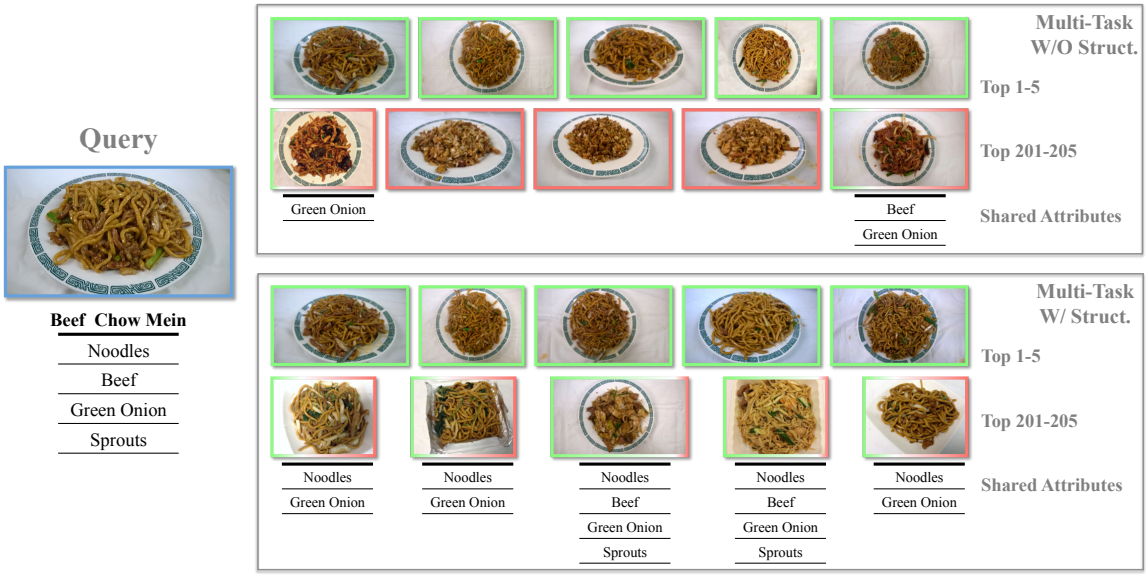


Figure 3.14: Retrieved images in the fine-grained food dataset. Green means the same fine-grained category, green-red means different fine-grained but the same coarse category (the ratio between green and red indicates similarity scores, i.e., Jaccard similarity), and red means sharing no attributes.

CHAPTER 4: LEARNING FINE-GRAINED FEATURE REPRESENTATION INVARIANT TO IRRELEVANT FACTORS

4.1 Motivation

In the previous chapter, we discussed how to embed hierarchical and shared attributes information to learn better feature representation. However, there are lots of factors that are not relevant to the categorization. Fig. 4.1 shows several examples in face recognition task. Attributes like illumination, viewpoint, hair style are entirely irrelevant to identification. But they may cause huge difference on the images of the same person. Therefore, we propose a method based on triplet network to learn features that are invariant to those type of attributes. This approach is orthogonal to the method described in the previous chapter and could be used together to boost the performance.

Figure 4.1: These images are from CelebA dataset [68]. Each column contains two images from the same person. They may look very different because of the illumination, viewpoint, hairstyle, facial expression, etc.

Ideally, images from the same person should look more similar than images from another person. In other words, features extracted from the images of the same person

should be closer than the features extracted from the different person. But that is not always the case because of those identification irrelevant attributes. For example, people with the same hairstyle may look more similar than the same individual who has a different hairstyle. So does the learned features. That is the central motivation for proposing a method that could learn features without those attributes information.

The main idea about proposed method is finding a way to measure samples in attribute space and try to push those who have the same attribute label but in different categories away from each other.

4.2 Methodology

In this section, we introduce our proposed framework of learning robust features that are invariant to the category irrelevant attributes. First, we describe the whole framework that contains a triplet network and attribute prediction network (if needed). Then, we demonstrate the generalized triplet loss in detail and discuss the sampling strategy.

4.2.1 Overview

Fig. 4.2 has two networks. The one in the bottom takes the original images and their attribute information as the input. Category labels are used in sampling procedure to build triplets and attribute information is used in the loss computing. Usually, not all datasets could have both class annotation and attribute annotation at the same time. A simple solution is to train an attribute prediction network with another dataset that has the attribute label in advance. Then we could send the target dataset to the pre-trained attribute network and use the output as the attribute label.

4.2.2 Attribute Prediction Network

We could choose several classical network architecture (such as AlexNet [53], VGGNet [95], GoogLeNet [100]) for predicting the attributes. Loss type (i.e. format of output $g(\cdot)$) is decided by annotation type. E.g., softmax loss for multi class label,

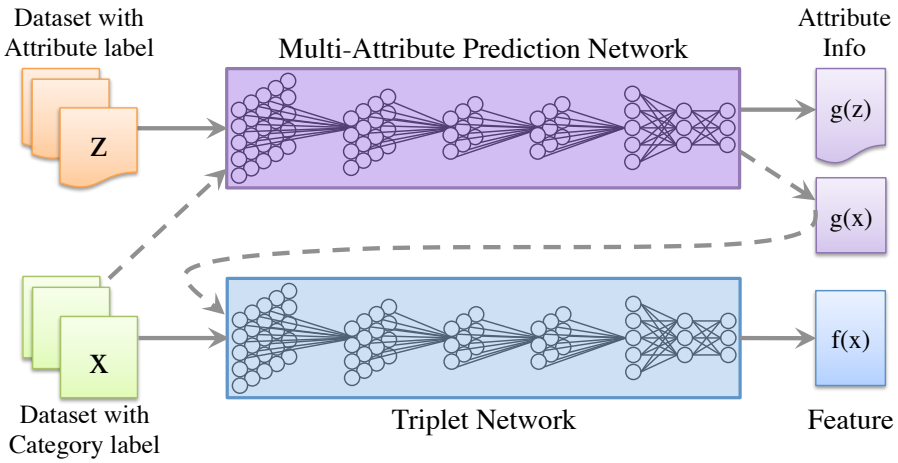


Figure 4.2: Dataset Z with desired attribute annotation is used to train a deep network that can predict attribute label $g(z)$. X is the dataset that we want to learn its feature representation $f(x)$ for. If it doesn't contain attribute annotation, we could use the output from attribute prediction network $g(x)$ instead. Original images X and their corresponding attribute information $g(x)$ are sent to the triplet network to generate the feature representation $f(x)$.

Euclidean loss for continues label, cross entropy loss for multiple binary label, etc.

Generalization is always a big concern in this type of transfer learning module. Since our main application is face representation learning, detection and alignment are standard preprocessing steps for every dataset. So the changing of datasets do not cause that much difference.

Also, we could train the attribute prediction network in multitasking fashion using both Z dataset and X dataset. For dataset Z, we could use the supervised loss (i.e. softmax loss, cross entropy loss, Euclidean loss) discussed before. For dataset X, we could use unsupervised loss like pseudo label [60]. Since the dataset X is also involved and contribute in training procedure, the learned network should be more suitable for dataset X.

4.2.3 Generalized Triplet Loss for Invariant Feature Learning

Since category irrelevant attributes are also a part of the information of the original image, if we train a deep neural network without considering the supervision of these

attribute information, the learned feature may contain a lot of irrelevant factors that do not contribute to the recognition task. In the worst case, samples belong to the different categories but have same attribute label could be very close in the learned feature space. Therefore, the basic idea of our proposed method is to push these type of samples away from each other. Here is the loss function of the proposed method:

$$E(r, p, n, m') = \frac{1}{2N} \sum_{i=1}^{N} \max\{0, \mathscr{D}(r_i, p_i) - \mathscr{D}(r_i, n_i) + m'\}. \tag{4.1}$$

Following the notations in the previous chapter, the only difference here is the margin part. Instead of using the fixed marigin, a dynamic margin is applied for each triplet. Formulation of this margin is very similar to the triplet loss itself except it is computed in the attribute space instead of the feature space. The new margin is defined as:

$$m' = m_b + \alpha(\max\{0, \mathscr{D}(r_i, p_i) - \mathscr{D}(r_i, n_i) + m_g\}). \tag{4.2}$$

where m_b is the base margin and could be set in the same way as the original triplet loss, $\mathscr{D}(\cdot, \cdot)$ is the distance in the attribute space, m_g is the margin for attribute space, α is used for balancing the base margin m_b and the distances.

By defining this new margin, if the negative and reference samples are closer in the attribute space than the positive and reference samples by a certain margin m_g (we could call them "hard triplet"), the margin used in triplet network should be increased. The new margin makes the hard triplet more difficult to satisfy the constraint during the training. In other words, it increases the importance of hard triplet implicitly.

m_g plays the similar role in attribute space as the m' in the feature space. But

the strategy of setting these two margins are entirely different. Feature space is an unknown space that we are trying to learn. However, we know data distribution in the attribute space, because it is already given by the attribute label or the prediction network. Therefore we could carefully design the m_g and α according to the data distribution to make it more reasonable.

There is another benefit of computing the dynamic margin in the attribute space. Since the pre-computed margins indicate the difficulty of the triplets, we can utilize them to build a proper sampling strategy. Basically, we want to train the easy triplets first and increasing the difficulty during the training. So that the whole model will converge to a relatively good point quickly, and also could handle the hard cases when finished. It is the same idea as doing hard negative mining in the last few epochs. But if we just sort triplets according to their difficulty (i.e. margin), it may be easily biased to some hard categories and lead to a local minimum. Thus will destroy the whole model in some time. However, we could solve it easily by mixing the hard triplets with the normal ones. To sum up, having a measurement about the difficulty of triplets is helpful for designing the sampling strategy.

4.3 Experiments

4.3.1 Evaluation of Synthetic Data

For proof of concept, we create a synthetic dataset by following formulation:

$$X = F(l_{id}) + \lambda G(l_{at}, \rho_{id,at}) + E(\sigma). \tag{4.3}$$

where $F(l_{id})$ is vector related to the category label (perpendicular to each other), $G(l_{at}, \rho_{id,at})$ is the attribute vector with the parameter of correleation between categories and attributes$\rho_{id,at}$, $E(\sigma)$ is the Gaussian noise defined by σ. λ is a hyper-parameter that controls the scale of attrubute vector G. Among these parameters,

$\rho_{id,at}$ is the most important one which defines the relationship between the category label and attributes. In implementation, if $\rho_{id_i,at_j} = 0.8$, it means 80% of the data in category i has been added by jth attribute vector and this attribute is likely to contribute to the category classification. While if $\rho_{id_i,at_j} = 0.5$, it means that they are irrelevant.

In the preliminary experiment, we create 10 categories and only import one attribute with correlation 0.5. The σ is set to 0.2 and λ is 0.6.

We evaluate our proposed method by measuring attribute prediction accuracy on the original data, features learned by traditional triplet network and features learned by our proposed method.

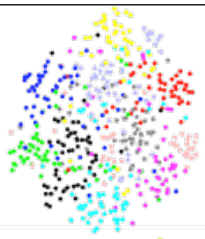
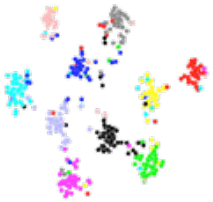
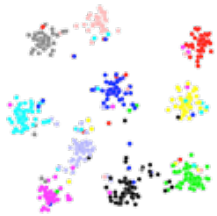
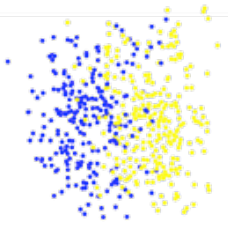
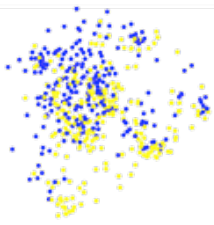
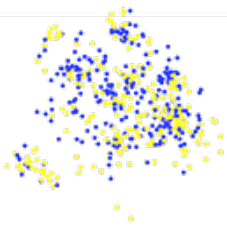
	Original Data	Traditional Triplet	Proposed Method
Attr Accuracy	85.86%	62.27%	56.40%
Plot by Category Label			
Plot by Attribute Label			

Figure 4.3: In the first row, we provide the attribute prediction accuracy. Then we plot the feature spaces according to the category label and attribute label in the second and third row.

Fig. 4.3 shows the experiment results. The prediction accuracy is decreasing, which indicates the information that could be used in attribute classification become less and less in learned features. When plotting by the category label, we could see that both traditional triplet and proposed method could provide enough discriminabil-

ity for classification. However, when we do the dimensionality reduction with linear discriminant model and color the data points with attribute label, the original data could show roughly two parts, features learned by traditional triplet network start overlapping, and in our proposed method, the yellow and blue dots are totally mixed. In other sentence, features of our proposed method contain the least attribute information.

4.3.2 Evaluation of Face Datasets

To evaluate our method in the real-world dataset, we employ Multi-PIE, a face recognition dataset [39] and select a subset of it to verify our idea of learning attribute invariant feature.

Multi-PIE dataset contains the images of the same person shot from different angles. We use frontal pose and a small portion of randomly selected non-frontal poses as the training set and create four testing sets with various level of difficulty (showed in Fig. 4.4).

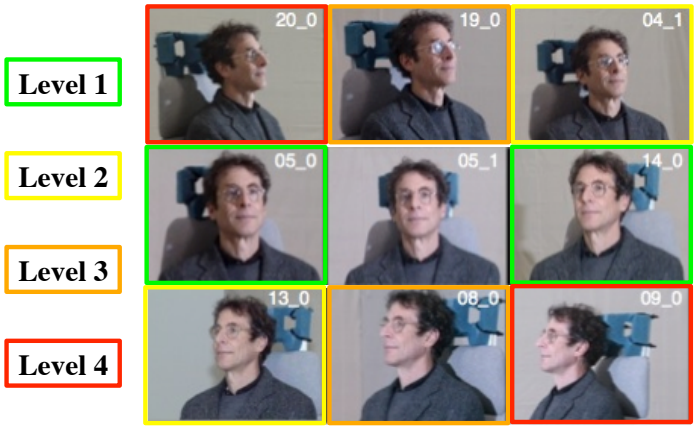


Figure 4.4: The difficulty is increasing from Level 1 to Level 4. Images in Level 1 testing set are the nearly frontal face. While images in Level 4 are shot from 90 degrees.

As shown in the table 4.1, our proposed method leads to a better recognition accuracy in all four levels. Because it contains less irrelvant information and more

Table 4.1: This table shows the recognition accuracy in four testing sets. The first row contains the results of traditional triplet network, and the second row shows the performance of our proposed method.

	Level 1	Level 2	Level 3	Level 4
Triplet	92.86	88.87	78.99	60.50
Ours	94.64	91.28	80.99	63.13

focus on the identification information.

4.4 Summary

In this chapter, we introduce a method of learning robust features that are invariant to the recognition irrelevant attributes. In our proposed method, attribute space is used to define the hardness of the triplet and involved in computing the dynamic margin. Such framework could provide higher classification accuracy by eliminating the irrelevant information which has been verified in the experiments using both synthetic data and face recognition data. This feature learning approach is sort of orthogonal to the method introduced in the previous chapter and could be used jointly for better results.

CHAPTER 5: LARGE-SCALE IMAGE INDEXING VIA SUPERVISED HASHING

5.1 Motivation

Given fine-grained feature representations extracted from large-scale databases, it is important to analyze them efficiently, e.g., in real time. Such analysis may include classification, categorization, segmentation, etc. In this chapter, we focus on content-based image retrieval (CBIR) [16, 114, 55, 108, 72, 79, 56, 66, 56, 109, 67, 46] , which has also been extensively investigated and applied in many applications, including medical image analysis [14, 144, 75, 1, 57]. Given an image database with labeled information, CBIR methods aim to retrieve and visualize images with feature representations most relevant to and consistent with the query image [61]. Note CBIR can also be used for classification purposes by considering the majority voting of the retrieved images. In this context, developing computational and scalable algorithms for large-scale image analysis is an urgent need when dealing with large databases.

In this chapter, we investigate hashing and binary coding methods for scalable retrieval, and we focus on medical image analysis as the use case to validate our algorithm. Particularly, we have built a scalable image-retrieval framework based on the supervised hashing technique and validate its performance on several thousand histopathological images acquired from breast microscopic tissues. Our method leverages a small amount of supervised information in learning to compress high-dimensional image feature vector into only tens of binary bits with the informative signatures preserved. The supervised information is employed to bridge the semantic gap between low-level image features and high-level diagnostic information, which is critical to medical image analysis. In the rest of this chapter, we introduce the overview of our image retrieval framework, and the details of the supervised hashing

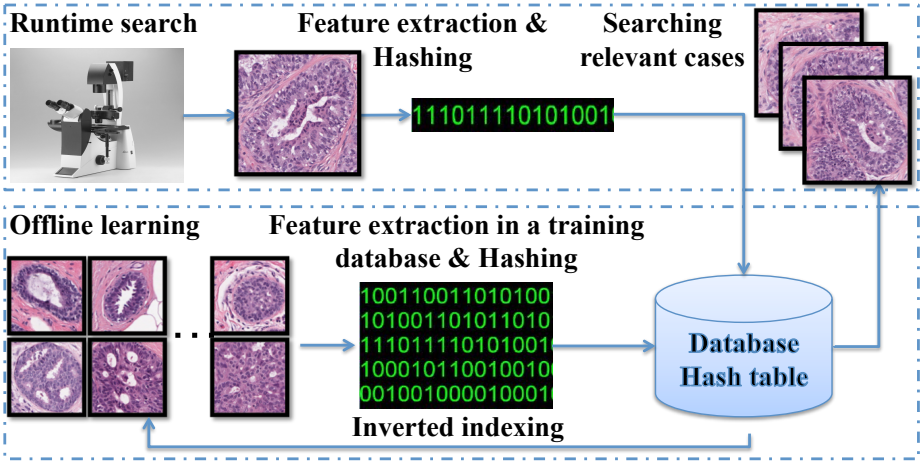


Figure 5.1: Framework of our large-scale image retrieval system. [138].

method. We also validate our framework in terms of both image classification and retrieval on a breast-lesion data set containing 3121 images from 116 patients and achieve an accuracy of 88.1% in a 10-ms query time for around 800 testing images and a precision of 83% in retrieval.

5.2 Methodology

5.2.1 Overview of Scalable Image Retrieval Framework

Fig. 5.1 shows a framework for the scalable image retrieval-based diagnosis system. It includes offline learning and run-time search. During the offline learning, we first extract high-dimensional visual features from digitized histopathological images. These features model texture and appearance information based on SIFT [70] and are quantized with a bag-of-words [96]. The SIFT descriptor is an effective local texture feature that uses the difference of Gaussian (DoG) detection result and considers the gradient of pixels around the detected region. It can provide an informative description of cell appearance and is robust to subtle changes in staining color. It has been used in both general computer vision tasks and histopathological image analysis.

Although these features can be used directly to measure the similarity among images, computational efficiency is an issue, especially when searching in a large database (e.g., exhaustively searching k-nearest neighbors). Therefore, we employ

a hashing method to compress these features into binary codes with tens of bits. Such short binary features allow easy mapping into a hash table for real-time search. Each feature is then linked to the corresponding training images using an inverted index. During a run-time query, high-dimensional features are extracted from the query image and then projected to the binary codes. With a hash table, searching for nearest neighbors can be achieved in a constant time, irrespective of the number of images. The retrieved images (via inverted indices of nearest neighbors) can be used to interpret this new case or for decision support based on majority voting.

5.2.2 Kernelized and Supervised Hashing

In this section, we introduce the key module for histopathological image retrieval, a kernelized and supervised hashing method.

Hashing Method: Given a set of image feature vectors $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ (in our case, $\boldsymbol{x}_i$ is the high-dimensional texture feature extracted from the ith histopathological image), a hashing method aims to find a group of proper hash functions h: $\mathbb{R}^d \mapsto \{1, -1\}^1$, each of which generates a single hash bit to preserve the similarity of original features. Searching k-nearest neighbors using tens of bits is significantly faster than traditional methods (e.g., Euclidean distance–based brute-force search), owing to constant-time hash-table lookups and/or efficient Hamming distance computation. Note that hashing methods are different from dimensionality-reduction techniques, since a fundamental requirement of hashing is to map similar feature vectors into the same bucket with high probability. Fig. 5.2 visualizes desirable hash functions as a hyperplane to separate higher-dimensional features. Therefore, hashing methods need to ensure that the generated hash bits have balanced and uncorrelated bit distributions, which leads to maximum information at each single bit and minimum redundancy among all bits.

Kernelized Hashing: Kernel methods can handle practical data that are mostly linearly inseparable. For histopathological images, linear inseparability is an impor-

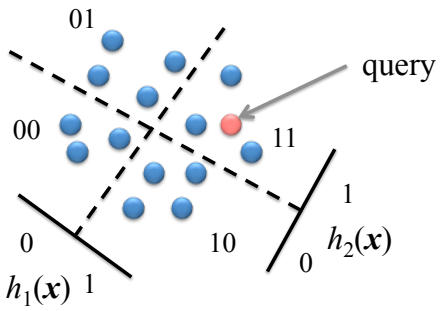


Figure 5.2: Visualization of desirable hash functions as a hyperplane.

tant constraint that needs to be taken into account when building hashing methods. Therefore, kernel functions should be considered in hashing methods $h = sgn(f(x))$ [56] to map the feature vectors into higher-dimensional space. A kernel function is denoted as κ: $\mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$. The prediction function f: $\mathbb{R}^d \mapsto \mathbb{R}$ with kernel κ plugged in is defined as

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \kappa(\boldsymbol{x}_{(j)}, \boldsymbol{x}) a_j - b, \tag{5.1}$$

where $\boldsymbol{x}_{(1)}, \cdots, \boldsymbol{x}_{(m)}$ are $m (m \ll n)$ feature vectors randomly selected from $\mathcal{X}$, $a_j \in \mathbb{R}$ is the coefficient, and $b \in \mathbb{R}$ is the bias.

The bits generated from hash functions h using f aim to keep as much information as possible, so the hash functions should produce a balanced distribution of bits, i.e., $\sum_{i=1}^{n} h(\boldsymbol{x}_i) = 0$. Therefore, b is set as the median of $\{\sum_{j=1}^{m} \kappa(\boldsymbol{x}_{(j)}, \boldsymbol{x}_i) a_j\}_{i=1}^{n}$, which is usually approximated by the mean. Adding this constraint into Eq. 5.1, we obtain

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \left(\kappa(\boldsymbol{x}_{(j)}, \boldsymbol{x}) - \frac{1}{n} \sum_{i=1}^{n} \kappa(\boldsymbol{x}_{(j)}, \boldsymbol{x}_i) \right) a_j = \boldsymbol{a}^\top \bar{\boldsymbol{k}}(\boldsymbol{x}), \tag{5.2}$$

where $\boldsymbol{a} = [a_1, a_2, ..., a_m]^\top$. $\bar{\boldsymbol{k}} : \mathbb{R}^d \mapsto \mathbb{R}^m$ is $\bar{\boldsymbol{k}}(\boldsymbol{x}) = \left[\kappa(\boldsymbol{x}_{(1)}, \boldsymbol{x}) - \mu_1, \cdots, \kappa(\boldsymbol{x}_{(m)}, \boldsymbol{x}) - \mu_m\right]^\top$, in which $\mu_j = \sum_{i=1}^{n} \kappa(\boldsymbol{x}_{(j)}, \boldsymbol{x}_i)/n$.

The vector $\boldsymbol{a}$ is the most important factor that determines hash functions. In traditional kernelized hashing methods, $\boldsymbol{a}$ is defined as a random direction drawn from a

Gaussian distribution [56], without using any other prior knowledge (i.e., no semantic information). This scheme works well for natural images, especially scenes, because of large differences in their appearance. However, such differences are very subtle in histopathological images. For example, identifying subtle differences between benign and actionable categories may require characterizing cytoplasmic texture or nuclear appearance. This subtlety motivates us to leverage supervised information to design discriminative hash functions that are suitable for histopathological image retrieval.

Supervised Hashing: Intuitively, hashing methods minimize the Hamming distance of "neighboring" image pairs (e.g., close in terms of the Euclidean distance in the raw feature space). "Neighboring" in our case is defined by its semantic meaning, i.e., whether the two images belong to same category or not. Therefore, supervised information can be naturally encoded as similar and dissimilar pairs. Specifically, we assign the label 1 to image pairs when both are benign or actionable, and -1 to pairs when one is benign and the other is actionable (as shown in Fig. 5.3). Then, l ($l \ll n$) feature vectors are randomly selected from $\mathcal{X}$ to build the label matrix S. Note that we need to provide labels for only a small number of image pairs. Therefore, labeled data are explicitly constrained by both semantic information and visual similarities, whereas unlabeled data are mainly constrained by visual similarities and implicitly affected by labeled data.

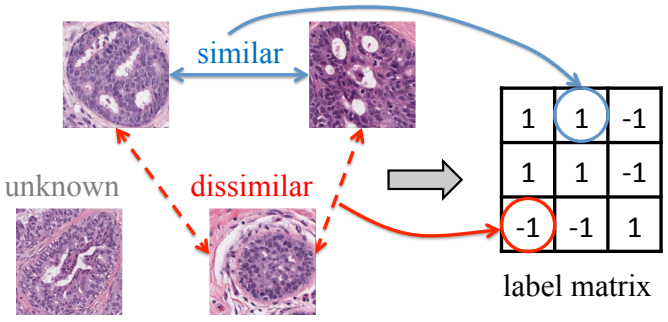


Figure 5.3: Supervised information is encoded in the label matrix S.

Using this supervision scheme to bridge the semantic gap, r hash functions $h_k(\boldsymbol{x})_{k=1}^{r}$

are then designed to generate r discriminative hash bits based on Hamming distances. However, direct optimization of the following Hamming distances $\mathcal{D}_h(\boldsymbol{x}_i, \boldsymbol{x}_j) = |\{k|h_k(\boldsymbol{x}_i) \neq h_k(\boldsymbol{x}_j), 1 \leq k \leq r\}|$ is nontrivial. Therefore, code inner products can be used to simplify the optimization process. As shown in [66], a Hamming distance and a code inner product are actually equivalent.

$$code_r(\boldsymbol{x}_i) \circ code_r(\boldsymbol{x}_j) = r - 2\mathcal{D}_h(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{5.3}$$

where $code_r(\boldsymbol{x})$ are r-bit hash codes and the symbol $\circ$ is the code inner product.

Therefore, the objective function $\mathcal{Q}$ to the binary codes H_l is defined as

$$\min_{H_l \in \{1,-1\}^{l \times r}} \mathcal{Q} = \left\| \frac{1}{r} H_l H_l^\top - S \right\|_F^2, \tag{5.4}$$

where $H_l = \begin{bmatrix} h_1(\boldsymbol{x}_1), \cdots, h_r(\boldsymbol{x}_1) \\ \cdots\cdots \\ h_1(\boldsymbol{x}_l), \cdots, h_r(\boldsymbol{x}_l) \end{bmatrix}$ is the the code matrix of the labeled data $\mathcal{X}_l$ and S is a label matrix with 1 for similar pairs and -1 for dissimilar pairs. $\|.\|_F$ denotes the Frobenius norm. Define $\bar{K}_l$ as $[\bar{\boldsymbol{k}}(\boldsymbol{x}_1), \cdots, \bar{\boldsymbol{k}}(\boldsymbol{x}_l)]^\top \in \mathbb{R}^{l \times m}$, $\bar{\boldsymbol{k}}(\boldsymbol{x}_i)$. The inner product of code matrix H_l can be represented as $H_l H_l^\top = \sum_{k=1}^{r} sgn(\bar{K}_l \boldsymbol{a}_k)(sgn(\bar{K}_l \boldsymbol{a}_k))^\top$ for binarization. Therefore, the new objective function $\mathcal{Q}$ that offers a clearer connection and easier access to the model parameter $\boldsymbol{a}_k$ is

$$\min_{\boldsymbol{a}_k} \quad \mathcal{Q}(\boldsymbol{a}_k) = \left\| \sum_{k=1}^{r} sgn(\bar{K}_l \boldsymbol{a}_k)(sgn(\bar{K}_l \boldsymbol{a}_k))^\top - rS \right\|_F^2 \tag{5.5}$$

This can be optimized using 1) spectral relaxation [114] to drop the sign functions and hence convexify the object function, or 2) sigmoid smoothing to replace $sgn()$ with the sigmoid-shaped function. In our implementation, we employ the first strategy to efficiently obtain a solution as the initialization, and use the second strategy to

produce an accurate solution.

5.3 Experiments

5.3.1 Data Description

Breast-tissue specimens available for this study were collected on a retrospective basis from the IU Health Pathology Lab (IUHPL) according to the protocol approved by the Institutional Review Board (IRB) for this study. All the slides were imaged using a ScanScope® digitizer (Aperio, Vista, CA) available in the tissue archival service at IUHPL. 3121 images (around 2250 K pixels) were sampled from 657 larger region-of-interest images (e.g., 5K×7K) of microscopic breast tissue, which were gathered from 116 patients. 53 of these patients were labeled as benign (usual ductal hyperplasia (UDH)) and 63 as actionable (atypical ductal hyperplasia (ADH) and ductal carcinoma in situ (DCIS)), based on the majority diagnosis of nine board-certified pathologists. To demonstrate the efficiency of our method, one fourth of all patients in each category were randomly selected as the test set and the remainder used for training. Note that each patient may have different number of images. Therefore, the number of testing images is not fixed. The approximate number is about 700-900 in each testing process. All the experiments were conducted on a 3.40 GHz CPU with 4 cores and 16G RAM, in a MATLAB implementation.

In each image, 1500 to 2000 SIFT descriptors were extracted from key points detected by DoG [70]. These descriptors were quantized into sets of cluster centers using bag-of-words, in which the feature dimension equals the number of clusters. Specifically, we quantize them into high-dimensional feature vectors of length 10,000, to maximally utilize these millions of cell-level texture features. We provide both qualitative and quantitative evaluations for our proposed framework on two tasks, image classification (i.e., benign vs. actionable category) and image retrieval, in terms of accuracy and computational efficiency.

5.3.2 Evaluation of Image Classification

In our system, classification is achieved using the majority vote of the top images retrieved by hashing. We compare our approach with various classifiers that have been widely used in systems for histopathological image analysis. Specifically, kNN has often been used as the baseline in analyzing histopathological images [101, 122], owing to its simplicity and proved lower bound, despite the inefficiency in large-scale databases. The Bayesian method is another solution to ensemble statistics of all extracted features and minimize the classification metric, which shows its efficacy in classifying histopathological images [14]. Boosting methods are always employed to combine multiple weak classifiers for higher accuracy [122, 22]. SVM with a non-linear kernel is commonly used in histopathological images because of its efficiency and the ability to handle linearly inseparable cases [102, 9, 77, 42]. For fair comparison, all parameters of these compared methods were optimized by cross-validation.

In addition, we also compared our proposed method with several dimensionality-reduction algorithms in terms of classification accuracy. Principal component analysis (PCA) has been widely used in this area to preserve variance of original features [92]. Graph embedding is a non-linear dimensionality-reduction algorithm that performs well in grading of lymphocytic infiltration in HER2+ breast cancer histopathology [3]. Since we use supervised information in generating hash functions, a supervised dimensionality reduction algorithm, neighborhood components analysis (NCA) [35], was also chosen for our experimental comparisons.

Fig. 5.4 shows the quantitative results for the classification accuracy. Most methods achieve better accuracy with higher-dimensional features. This is very intuitive, as finer quantization of SIFT features usually provides richer information. In particular, since the SIFT interest points cover most nuclear regions in images, fine quantization (i.e., high-dimensional features) indicates analysis on a small scale. Exceptions are the Adaboost and Bayestian methods, whose accuracy drops when the feature

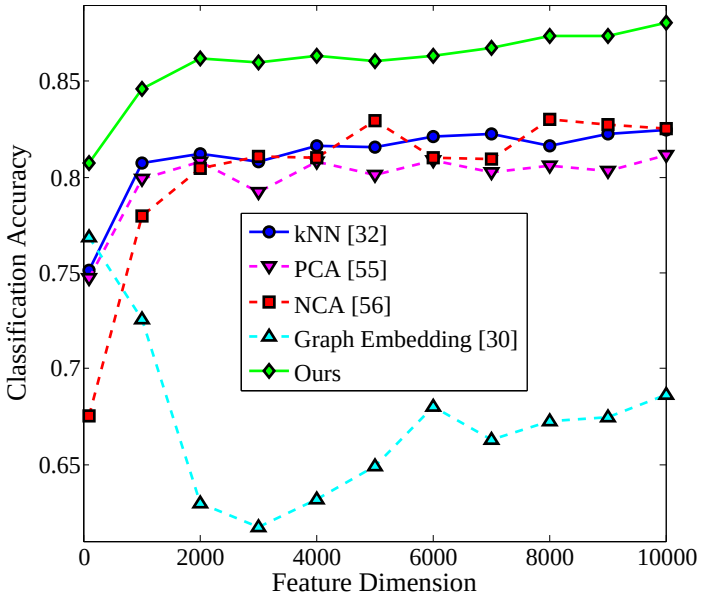


Figure 5.4: Comparison of classification accuracy with different dimensions of features (from 100 to 10000).

dimensions increase. This indicates that high-dimensional features do not guarantee the improvement of accuracy. An important factor is the proper utilization of such information. For example, Adaboost is essentially a feature-selection method that chooses only an effective subset of features for the classification. Therefore, it may lose important information, especially in high-dimensional space, resulting in accuracy worse than that of our hashing method. Our method is also generally better than kNN and its variations, owing to the semantic information (i.e., labels of similar and dissimilar pairs in hashing) that bridges the semantic gap between images and diagnoses. Note that our hashing method needs only a small amount of supervision – in this case, similar or dissimilar pairs of 40% images. This is generally less than the supervised information required by SVM in the training stage. It compares favorably to all other methods when the feature dimension is larger than 1000. The overall classification accuracy is 88.1% for $10,000$-dimensional features, 2% to 18% better than other methods.

Fig. 5.5 compares the computational efficiency of these methods. With increasing dimensionality the running time of some compared methods increases dramatically.

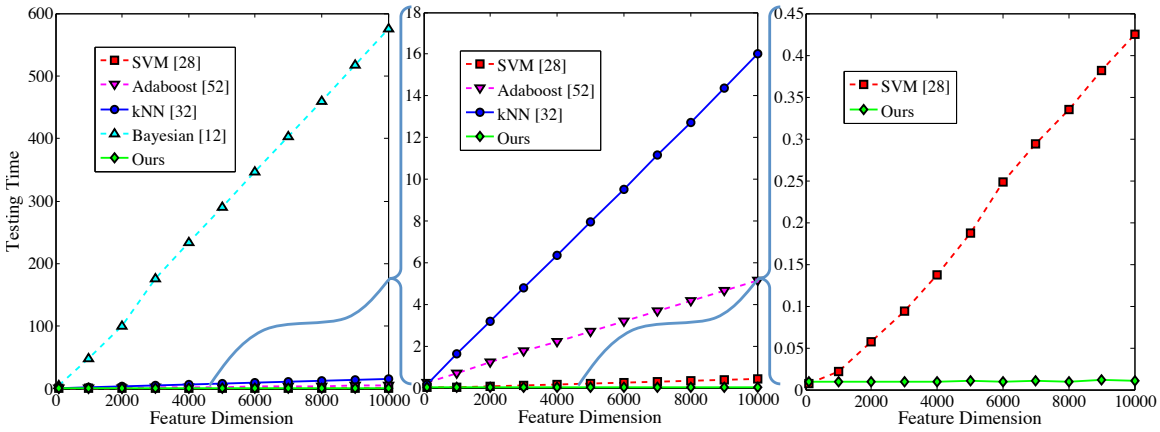


Figure 5.5: Comparison of the classification running time (seconds) with different dimensions of features, which means the average time of classifying hundreds of test images.

When feature dimensionality reaches $10,000$, kNN needs 16 seconds to classify all query images, and Adaboost needs 5 seconds. SVM, dimensionality-reduction methods, and the proposed method are much faster. However, the running time for SVM increases with the feature dimensionality, as shown in the expanded view of Fig. 5.5. In contrast, PCA, graph embedding, NCA, and ours achieve constant running time in this data set owing to the fixed size of features after compression. Compared to other dimensionality-reduction methods, our approach is about 10 times faster because of the efficient comparison among binary codes. In addition, the running time of all kNN-based methods increases with the number of images in a data set, as exhaustive search is needed, while hashing-based methods can achieve $\mathcal{O}(1)$ efficiency using a hash table. To summarize, the average running time of our method is merely 0.01 second for all testing images, which is 40 times faster than SVM and 1500 times faster than kNN.

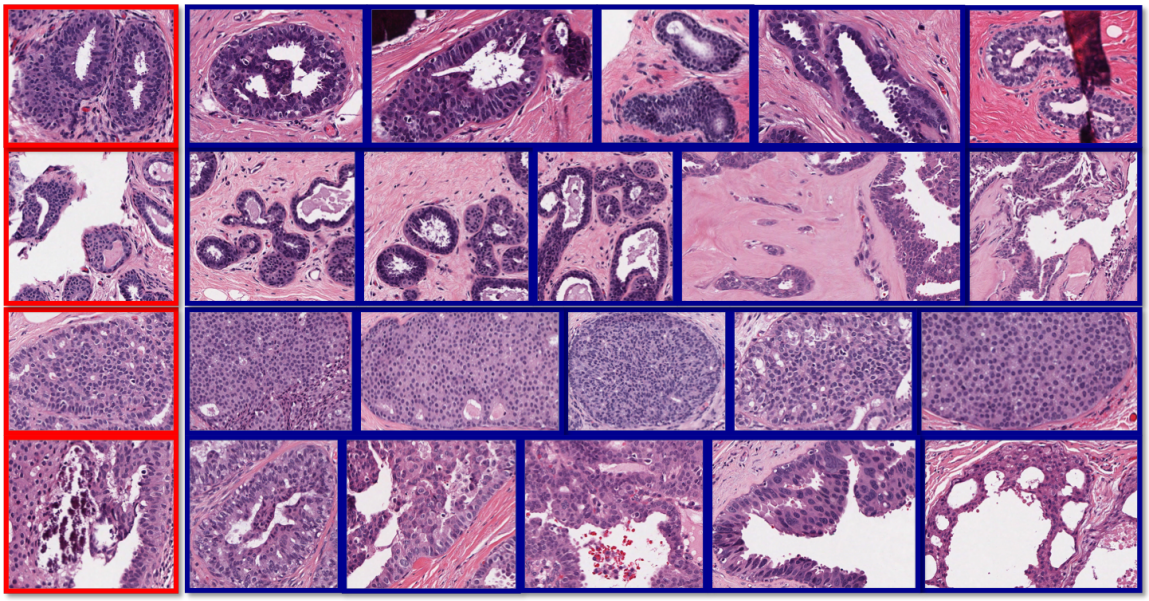
5.3.3 Evaluation of Image Retrieval

We have also conducted experiments on image retrieval using $10,000$-dimensional features. The retrieval precision, evaluated at a given cut-off rank and considering only the topmost results, is reported in Table 5.1, along with the query time and memory cost. The results are quite consistent with the image classification. The

Table 5.1: Comparison of retrieval precision for the top 10, 20, and 30 results (denoted as P@10, P@20 and P@30, respectively), along with the memory cost of training data and query time of all test images. Both mean values and the standard deviation (STD) of 20 experiments are reported. The best precision in each row for benign and actionable categories are highlighted.

	kNN		**PCA**		**NCA**		**Graph Embedding**		**Ours**	
	benign	actionable	benign	actionable	benign	actionable	benign	actionable	benign	actionable
P@10	0.779	0.687	0.762	0.705	0.799	0.697	0.672	0.487	**0.836**	**0.830**
P@20	0.773	0.653	0.758	0.681	0.800	0.689	0.673	0.486	**0.839**	**0.829**
P@30	0.770	0.631	0.755	0.667	0.800	0.685	0.670	0.480	**0.837**	**0.833**
STD	0.024		0.028		0.020		0.012		0.011	
Time (s)	15.77		10.07		10.04		10.03		<0.01	
Memory	134.58MB		0.65MB		0.65MB		0.65MB		0.01MB	

mean precision of the hashing method is around 83%, and the standard deviation is 1.1%, which is much better than PCA [92], graph embedding [3] and NCA [35]. In most cases, the precision of our method is at least 6% better than the others, except the NCA. Our method is around 3.5% better than NCA on benign cases. To demonstrate statistical significance, we perform *t-test* for the precision obtained by NCA and by the proposed method on benign cases, under the null hypothesis using a significance level of 0.05. The *p-values* are found as 3.6×10^{-6}, 3.2×10^{-6} and 5.7×10^{-6} at the range of top 10, 20 and 30 retrievals, respectively, demonstrating that precision values achieved by the proposed technique are indeed significantly better than NCA on the benign cases. In addition, our method is around 14% better than NCA in the actionable cases, resulting much higher average precision. In fact, most traditional methods produce such highly unbalanced results as NCA does, i.e., the retrieval precision of the benign category is much higher than that of the actionable one. In contrast, our method does not have this problem, owing to the supervised information and the optimization for balanced hash bits. Our framework is also computationally more efficient than traditional methods. The query time of our hashing method is a thousand times faster than kNN and ten times faster than other dimensionality-reduction methods. Note that our method takes a constant time when using the hash table, independent of the number of feature dimensions and the number of samples.



(a) Query (b) Retrieved Images

Figure 5.6: Four examples of our image retrieval (query marked in red and in the first column, and retrieved images marked in blue). The first two rows are benign; the last two rows are actionable.

Furthermore, the memory cost is also considerably reduced ($10,000$ times less than that of kNN). Therefore, this method is more applicable to large-scale databases (millions of images) than are other methods.

Fig. 5.6 shows our image-retrieval results. The top five relevant images are listed for each query image. The differences between certain images in different categories are very subtle. Our accurate results demonstrate the efficacy of the proposed method. Specifically, the features capturing local texture and appearance are very robust to various image sizes, cell distributions, and occlusions by the blood. The supervised information also improves the retrieval precision by correlating binary code with diagnosis information. These retrieved images are clinically relevant in potential (i.e., retrieved images belong to the same category as the query image) and thus can be useful for decision support.

5.3.4 Discussions

We discuss the benefits of the algorithm, parameter sensitivity, implementation issues, and limitations here.

Regarding the choice of high-dimensional features, around 1000 dimensions have usually been used for quantization by many previous studies, a number that has been proved to achieve good accuracy. Using lower-dimensional features (e.g., 100) is not accurate, while using higher-dimensional features is not efficient, and the improvement of accuracy could be marginal. This is consistent with our experimental results shown in Fig. 5.4, i.e., a performance jump from 100 to 1000 dimensions. On the other hand, when analyzing histopathological images, using high-dimensional features (e.g., 10,000) implies nearly cell-level analysis, which is actually beneficial for the accuracy, even though the accuracy gain is not as big as jumping from 100 to 1000. Therefore, we have introduced hashing methods to harvest the benefits of high-dimensional features, without sacrificing computational efficiency.

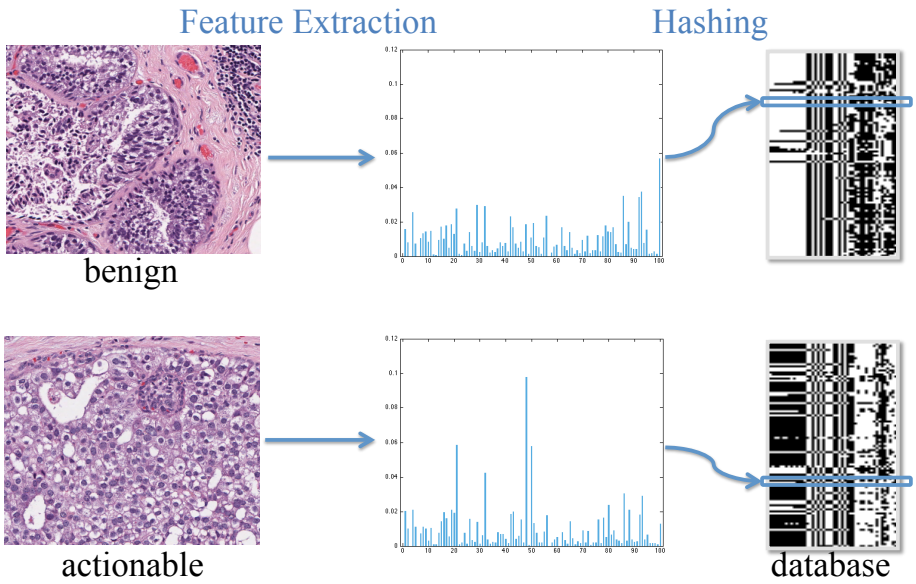


Figure 5.7: Visualization of compressed hash bits. Their distribution well separates the begin and actionable categories.

Regarding supervised information, it significantly improves classification accuracy

thanks to the discriminative modeling of the hashing function in an attempt to bridge the semantic gap. In Fig. 5.7, we randomly selected 100 samples from benign and actionable categories and visualized their 48 hash bits. The distributions of hash bits are clearly different between the two categories, explaining the high accuracy for classification. We also quantitatively investigated the benefits of using supervised information. Specifically, we evaluated our method when using 10% to 100% supervision or training labels, as shown in Fig. 5.8. The gain in accuracy is very high (from 71% to nearly 87%) when the ratio of training labels increases from 10% to 40%, which demonstrates the efficacy of using supervised information. For more than 40% labels, the improvement of accuracy becomes marginal, reaching 88% accuracy when using 100% labels. This means that our method needs only a small portion of labels to achieve high accuracy, owing to the unified framework of coupling Hamming distance optimization and supervised information.

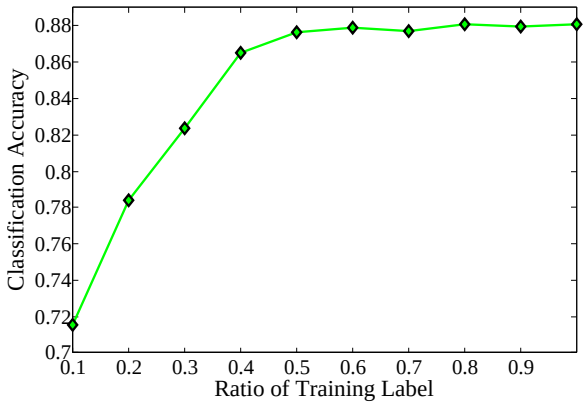


Figure 5.8: Classification accuracy when using 10% to 100% supervision.

One of the most significant benefits of our proposed framework is the computational and storage efficiency. Comparing 48 bits with Hamming distance or hash table is substantially faster than using high-dimensional features. However, a natural question is whether the length of hashing bits affects the accuracy and retrieval precision. Therefore, we evaluated the effect of hashing-bit lengths ranging from 1 to 48. Theoretically, 1 bit is sufficient for binary classification purpose, i.e., actionable vs. benign.

In fact, as shown in Fig. 5.9, using 8 bits already achieves high accuracy for classification. However, such short code is not discriminative enough for image retrieval. For example, 8 bits can represent only 64 hash values. This means that nearly 50 images are mapped into the same hash value, which is an unordered list with zero Hamming distance. Retrieving them may not be beneficial for decision support. On the other hand, using more than 64 bits adversely affects computational efficiency, since the hash table is no longer an option owing to memory constraint. Therefore, we chose 48 bits for this task, ensuring sound accuracy for classification and high relevance for retrieval without sacrificing efficiency. We expect that our scalable framework can be efficiently used for real-time querying of very large databases.

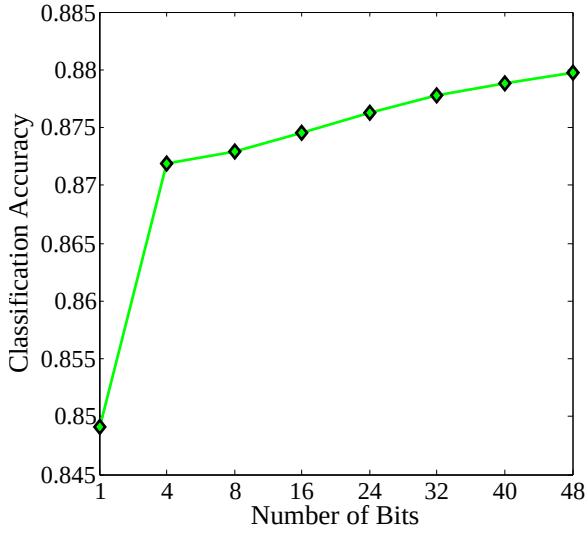


Figure 5.9: Classification accuracy with different lengths of hashing bits.

In the task of image retrieval, our method effectively retrieves images with morphological and architectural image patterns similar to the query image, as shown in Fig. 5.6. This can be explained by the capability of the hashing function in leveraging both diagnostic information and visual similarities. In other words, hash bits can simultaneously encode local textural features with semantic labels.

5.4 Summary

In this chapter, we introduce a *scalable image-retrieval framework* for intelligent histopathological image analysis. Specifically, we employed hashing to achieve efficient image retrieval and presented a kernelized and supervised hashing approach for real-time image retrieval. The potential applications of our framework include image-guided diagnosis, decision support, education, and efficient data management.

CHAPTER 6: LARGE-SCALE IMAGE INDEXING VIA WEIGHTED HASHING

6.1 Motivation

In the previous chapter, we introduce the supervised hashing for histopathological image retrieval, by indexing high-dimensional features with binary codes. The high-dimensional features approximately represent cell-level information, while it is still different from exhausively analyzing each individual cell. Such thorough examination is necessary in many use cases. Take the lung histopathological image analysis as an example, it is important to differentiate the adenocarcinoma and squamous carcinoma, both of which belong to the non-small cell carcinoma. The main challenge of this task is the need of analyzing all individual cells for accurate diagnosis, since the difference between the adenocarcinoma and squamous carcinoma highly depends on the cell-level information, such as its morphology, shape and appearance. Although rigorously measuring and analyzing each individual cell is important and can assist pathologists for accurate diagnosis, a region-of-interest (ROI) image may contain hundreds or thousands of cells, and analyzing each cell is computationally inefficient using traditional methods, if not infeasible. Using cell segmentation and cell retrieval via hashing offers a potential solution, i.e., designing an automatic framework for the large-scale cell-level analysis of histopathological images, which can segment and retrieve millions of cells in real-time by hashing. However, due to the imperfect segmentation methods and several inherent limitations of hashing methods, directly applying hashing methods, including supervised hashing, may have issues. Therefore, we propose to improve the traditional hashing methods by incorporate weights, to emphasize important hash values. Based on this imporvement, we conduct extensive experiments to differentiate adenocarcinoma and squamous carcinoma, using a large

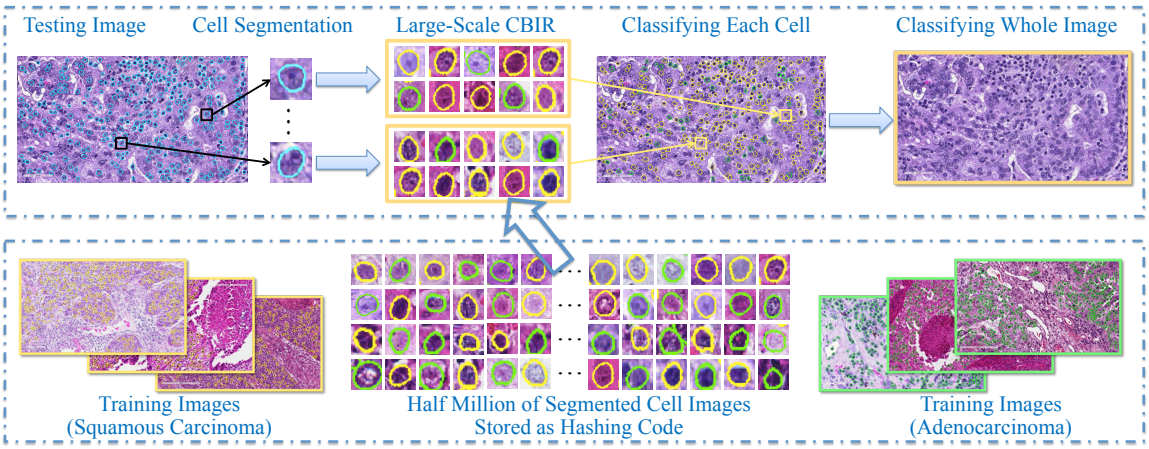


Figure 6.1: Overview of our proposed framework, based on robust cell segmentation and large-scale cell image retrieval. The top row is the online classification, and the bottom row is the offline learning. Yellow boundaries mean squamous carcinoma, green means adenocarcinoma, and blue means unknown types to be classified.

dataset containing thousands of lung microscopic tissue images acquired from hundreds of patients. Our proposed framework achieves 87.3% accuracy in real-time, by retrieving a massive database of half million cells extracted from this dataset.

In the rest of this chapter, we first introduce overview of our framework for cell-level histopathological image analysis. Then, we elaborate the limitations of using traditional hashing methods for this cell retrieval task, including the supervised and kernelized hashing [66]. After that, we elaborate the details of our improvements by incorporating weights into the hashing framework. We also provide thorough evaluations and comparisons of our method in the experiment section.

6.2 Methodology

6.2.1 Overview

Fig. 6.1 shows the overview of our proposed framework, which includes offline learning and online classification. During offline learning, our system automatically detects and segments all cells from thousands of images, resulting in half million of cell images. Regarding cell detection and segmentation, we employ the single-pass voting

(SPV) scheme [84, 119][1]. After that, texture and appearance features are extracted from these cell images and are compressed as binary codes, i.e., tens of bits. These compressed features are stored in hash table for constant-time access even among millions of images.

During online classification, our system segments all cells from a testing image, and same types of features are extracted accordingly and compressed using hashing methods. Then, we perform large-scale cell image retrieval for each segmented cell to classify its category. Finally, the classification result of the testing image is decided by the majority logic, i.e., voting from all cells' classification. Using this scheme, our system can maximally utilize the cell-level information without sacrificing the computational efficiency, owing to the large-scale retrieval via hashing methods. We also design a content-aware weighting scheme to improve the accuracy of traditional hashing methods, based on the observations and priors in histopathological image analysis. In the following sections, we introduce the details of large-scale cell image retrieval and weighting techniques.

6.2.2 Hashing with Content-Aware Weighting

Given all cells that are segmented from a testing image, our system conducts cell-level classification by exhaustively comparing each cell with all cells in the training database, using hashing-based large-scale image retrieval and majority voting. Theoretically, using hashing methods by indexing in a hash table enables constant-time searching, no matter how many training samples are used. However, it also requires that the length of the binary code is sufficiently short, to store in physical memory for fast access. Given limited number of hash bits, an inevitable limitation is that a large number of images may be mapped into the same hash value. In other words, it may result in an unordered set for the same hash value, where exact or near-exact

[1]Cell detection and segmentation is not a focus of this dissertation, so we do not provide details. Please refer to [120].

matches may be obscured within a large-scale database due to noisy features, similar instances, or erroneous segmentations. This is particularly true for histopathological image analysis, since the differences of cells are very subtle, and accurate segmentation for all cells is challenging. Consequently, the accuracy of cell classification is adversely affected when choosing the majority of cells mapped into a hash value, and the accuracy of whole image classification is also reduced. Fig. 6.2 illustrates this inherent limitation of hashing methods in analyzing histopathological images. Half million of cells are mapped into 12 bits, which mean $2^{12} = 4096$ hash values. The entries (i.e., hash values) in each hash table are illustrated according to the distribution of cells mapped into them, such as the ratio between two categories (i.e., adenocarcinoma and squamous carcinoma) and the number of cells mapped into that entry. Ideally, each hash value should be discriminative enough, i.e., the number of one type should dominate the other. However, many of them actually contains similar amount of both types of cells, i.e., around 0.5 ratio. In other words, the indecisive hash values are usually around the 0.5 ratio, indicating equal opportunity for either category. Classification based on such hash value is likely inaccurate. The small circles in Fig. 6.2 are also not reliable, since only few cells are mapped there, which can be easily affected by the image noise or erroneous segmentation. A potential solution is to identify reliable hash values and omit indecisive one, by heuristically select or prune them via feature selection. However, this may involve tuning parameters and lack the consistent measures. Furthermore, there is no guarantee that the hash values from feature selection algorithms are sufficiently discriminative for classification.

Therefore, we introduce a probabilistic-based formulation to solve these problems in a principled way, i.e., design a content-aware weighting scheme to re-weight the importance of hash values. Specifically, we aim to assign probability scores to each hash value, based on its ability to differentiate different categories. Such “soft assignment" upon hash values can significantly boost the classification accuracy using hashing-

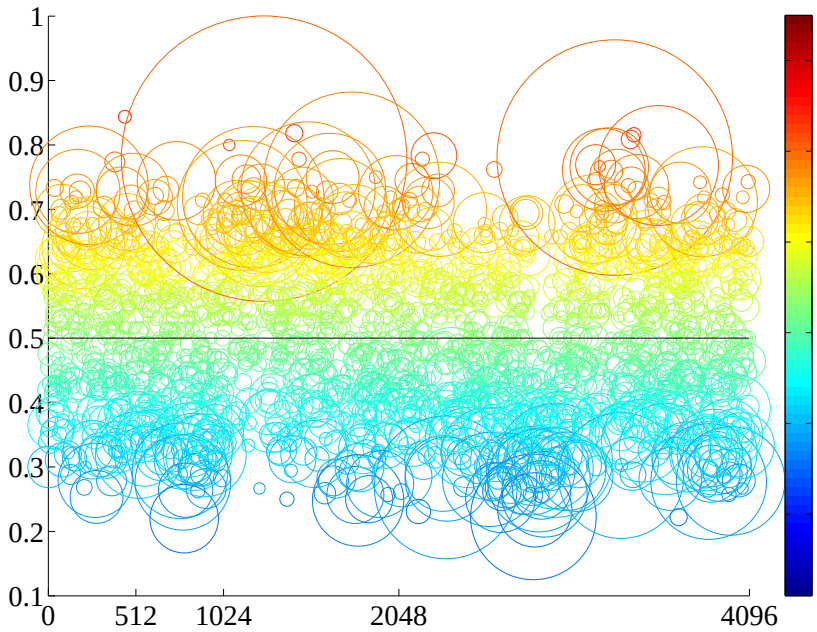


Figure 6.2: Illustration of the cell distribution in a hash table. X-axis means the hash value using 12 bits, ranging from 0 to 4095, and y-axis means the ratio between two types of cells, ranging from 0 to 1. Each circle means a set of cells mapped to the hash value located in the centroid, its size means the number of cells, and the color map visualizes the ratio of two types of cells, same as the y-axis values.

based retrieval. In our framework, kernelized and supervised hashing (KSH) [66] is employed as the baseline method to generate initial hash values, because of its efficacy and success in histopathological image analysis [137]. The content-aware weighting scheme can significantly enhance the differentiation ability of hash values generated by this baseline. Intuitively, since cells in certain hash values are not accurate for classification, their weights should be diminished during the process. On the other hand, discriminative hash values should be emphasized, e.g., circles nearby 1 or 0 ratios. In addition, small sizes of circles are not preferred and their weights should be reduced, as they can be easily affected by many factors such as unusual staining color, inaccurate segmentation results and image noise in our use case. Therefore, we designed two metrics to emphasize discriminative hash entries, with generalized notations for multi-class classification:

- Support: Given a specific hash value H, the number of cells mapped into H

should be considered. This indicates that such amount of cells are used for the classification of this hash value, each with contribution 1, while all remaining cells are irrelevant, i.e., contribution 0. Therefore, we name this metric as "support", which is conventionally referred to the set of numbers having non-zero values. Denote $S_H = \{\text{cell} : h(\text{cell}) = H\}$ as the set of cells mapping into a specific hash value H, where $h(\text{cell})$ is the hash value of the cell. The support W_H of the hash value H is defined as:

$$W_H = \frac{|S_H|}{\sum_{m=0}^{2^r-1} |S_m|} \tag{6.1}$$

where $|S|$ is the number of element in set S and r is the number of hash bits, representing 2^r hash values.

- Certainty: Instead of assigning a certain category label to each hash value, we should consider the confidence of such categorization and assign a probabilistic label to each hash value. Therefore, this "certainty" term defines the probability of a cell belonging to the ith category when its hash value is H:

$$\begin{aligned} P(L_i|H) &= \frac{P(L_i, H)}{P(H)} \\ &= \frac{|\{\text{cell} : l(\text{cell}) = L_i, \text{cell} \in S_H\}|}{|S_H|} \end{aligned} \tag{6.2}$$

where $l(\text{cell})$ is the label of a cell image and L_i means the ith label or category.

We combine these two weights to advocate the importance of highly discriminative hash values with sufficient support. Specifically, during the training process, W_H and $P(L_i|H)$ can be computed for all hash values. The category of a whole testing image is decided by:

$$\arg\max_i \sum_{\text{cell} \in \text{query}} W_{H_{\text{cell}}} P(L_i|H_{\text{cell}}) \tag{6.3}$$

where H_{cell} is the hash value of the cell belonging to the query (testing) image.

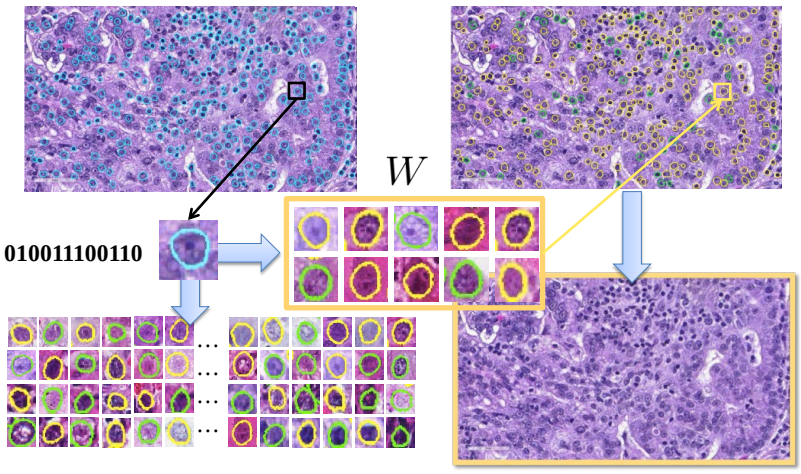


Figure 6.3: Workflow of the weighted hashing-based classification. Starting from an unknown image to be categorized, each segmented cell is classified by searching the most similar instances. Their results are combined via the content-aware weighting scheme, predicting the categorization for the whole image.

This content-aware weighting scheme effectively solves the issues of using hashing-based retrieval methods for classification. The importance of each cell is decided case-specifically, and accumulating the results of all cells provide accurate classification for the whole image. In addition, this framework is able to accommodate new samples efficiently. The updating scheme can be achieved by storing not only the weights but also the number of cells in each category. Given new samples, we can update the cell number in their mapped hash entries, re-calculate and update the weights based on such information. Regarding the computational complexity, the overhead during the testing stage lies in the weighted combination, which is negligible as demonstrated in the experiments. Therefore, this process is computationally efficient, same as traditional hashing methods. Fig. 6.3 summarizes the classification procedure using weighted hashing. The whole framework includes cell segmentation, hashing, and retrieval. The probability scores are assigned to each hash entry, and they are aggregated within the whole image for the final classification. Benefited from this thorough analysis of each individual cell, this framework can achieve promising accuracy without sacrificing the efficiency.

6.3 Experiments

6.3.1 Data Description

In this section, we conduct extensive experiments to evaluate our weighted hashing with multiple features for cell-level analysis. Our dataset is collected from the Cancer Genome Atlas (TCGA) [76], including 57 adenocarcinoma and 55 squamous carcinoma. 10 patches with 1712×952 resolution, i.e., region-of-interests (ROIs), are cropped from each whole slide scanned pathology specimens, by consulting with certified pathologists. Generally, the ROIs mainly consist of cancer cells. The lymphocytes regions which have different visual patterns than the representative tumor regions are avoided. All the data have been prepared and labeled based on the independent confirmation of the pathologists. In each image, our algorithm detects and segments around 430 cells. In total, 484,136 cells are used in to evaluate the system (195,467 adenocarcinoma cells and 288,669 squamous carcinoma cells). We evaluate the efficacy of our proposed framework in terms of the classification accuracy and computational efficiency. The evaluations are conducted on a 3.40GHz CPU with 4 cores and 16G RAM, in MATLAB and `C++` implementation.

6.3.2 Evaluation of Image Classification

In our framework, the image classification (i.e., differentiation of adenocarcinoma and squamous carcinoma) is conducted by examining all cells using hashing-based large-scale image retrieval with content-aware weighting. We compare our hashing-based classification scheme with several effective classifiers employed for histopathological image analysis. Following the convention, k-nearest neighbor (kNN) method is used as the baseline of analyzing histopathological images [101], owing to its simplicity and efficacy. Dimensionality reduction methods such as principal component analysis (PCA) are effective approaches to improve the computational efficiency and have been employed to analyze histopathological images using high-dimensional fea-

tures [92]. Support Vector Machine (SVM) is a supervised classification method and widely used in grading systems for breast and prostate cancer diagnosis [21]. We also compare with the traditional kernelized and supervised hashing (KSH) [66]. For fair comparison, same features are used for all compared methods, and their parameters and kernel selections are optimized by cross-validation. Specifically, we use an RBF kernel with optimized gamma value for SVM, and k=9 for kNN. Regarding dimensionality reduction, PCA compresses the original features (i.e., 144 dimensional texture feature base on Histogram of Oriented Gradients [15]) into 12 floats, and our hashing method generates 12 bits from each original feature.

Table 6.1: Quantitative comparisons of the classification accuracy (the mean value and standard deviation) and running time. Compared methods include kNN [101], PCA [92], SVM [21], KSH [66] and ours.

	Adeno	Squam	Average	Time(s)
kNN	0.309 ± 0.058	0.710 ± 0.072	0.514	2605.80
PCA	0.458 ± 0.084	0.954 ± 0.057	0.711	460.20
SVM	0.929 ± 0.085	0.704 ± 0.092	0.816	46.82
KSH	0.861 ± 0.076	0.763 ± 0.084	0.812	1.22
Ours	0.887 ± 0.069	0.854 ± 0.062	0.873	1.68

To conduct the comparison, we randomly select 20% patients as testing data (around 230 images, or $96,000$ cells), and use the images from remaining patients as training. This procedure is repeated for 30 times to obtain the mean and standard deviation. Table 6.1 shows the quantitative results of the classification accuracy. Despite the efficacy of kNN in many applications, it fails to produce reasonable results in this challenging problem, due to the large variance of cell images, noise in such large-scale database and unbalanced number of two classes. PCA reduces the feature dimensions, which could be redundancy information or noise. The classification accuracy is significantly improved, while still only around 70%. SVM incorporates supervised information, i.e., labels of adenocarcinoma and squamous carcinoma. Not

surprisingly, it largely outperforms unsupervised methods, with an accuracy of 81.6%. KSH has the same merit of using supervised information, and hence achieves comparable accuracy as SVM. Our proposed hashing method not only utilizes kernels and supervision, but also is equipped with the content-award weighting scheme to solve the inherent problems of hashing methods. Therefore, it outperforms all other methods, with an accuracy of 87.3%. In addition, the standard deviation of our algorithm is also relatively small, indicating the stableness of our algorithm. Table 6.1 also shows the individual accuracy of adenocarcinoma and squamous carcinoma. Besides the superior accuracy, our method also achieves the most balanced results for both cases, which is important to this clinical problem as both cases should be recognized and sacrificing the accuracy of one case is not acceptable.

Table 6.1 also compares the computational efficiency of these methods, i.e., the testing time for classification. Our hashing method compresses each feature into merely 12 bits, resulting in a hash table with 4096 values, which allow instant access to images mapped into any hash value. Therefore, KSH and our method is real-time, i.e., around 1-2 seconds. Our method uses content-aware weighting and is slightly slower than KSH, due to a small overhead for computing the weighted average. Such computational overhead (i.e., 0.4s) is negligible in practice. Other methods are all significantly slower, ranging from 46 to 2600 seconds. This is the main factor preventing previous methods from being used for cell-level analysis. Note that the detection and segmentation takes around tens of seconds for each image, and feature extraction takes half second, both of which are the same for all compared methods. The overall speed is quite efficient for practical use.

6.3.3 Discussions

In this section, we discuss the parameters, implementation issues and some limitations of our system, and their potential solutions.

Since the image classification relies on the features extracted from the segmented

cells, inaccurate segmentation may adversely affect the classification accuracy. Nonetheless, our system still generates accurate classification results, because of two reasons: 1) Most segmented cells are correct, which is reflected by the high precision and recall. 2) More importantly, the weighting scheme reduces the importance of unreliable features, most of which are extracted from inaccurate segmentations. Particularly, this weighting scheme ensures the robustness of the classification module, making it less sensitive to the segmentation precision. Therefore, our content-aware hashing method not only benefits the classification accuracy, but also is compatible with the paradigm of cell-level analysis, given the fact that most existing cell segmentation methods are still not perfect.

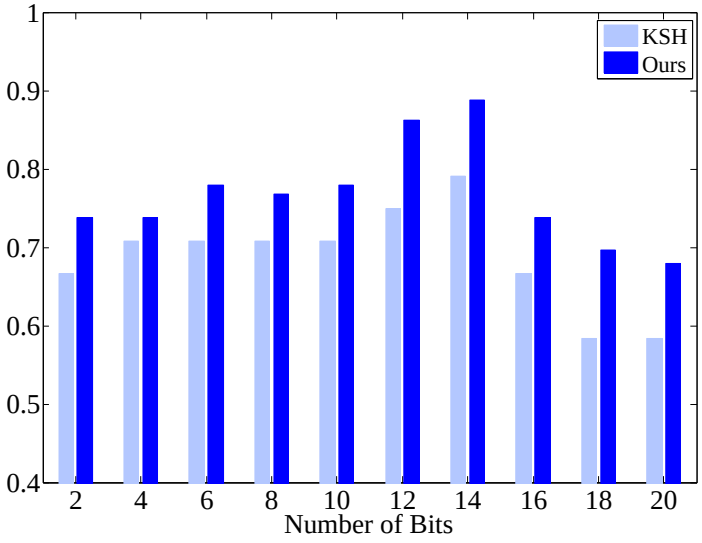


Figure 6.4: Classification accuracy of our content-aware hashing and KSH [66], using different number of hashing bits (2 to 20).

Our hashing-based classification has few parameters that are easy to choose and not sensitive. This is critical to an automatic framework for histopathological image analysis, since tuning sensitive parameters is infeasible when conducting this large-scale and cell-level analysis. Particularly, our hashing-based classification only has one parameter, i.e., the number of hash bits. In our experiments, we have used 12 bits for classification, indicating 4096 hash values. Theoretically, using one bit is already

sufficient for binary classification purpose, i.e., differentiation of two types of cells. However, as shown in Fig. 6.2, some hash values may not be reliable and have to be pruned, due to image noise and several inaccurate segmentations. Therefore, it is necessary to use many hash values, which also enable multi-label classification. On the other hand, it is also desired to have enough samples mapped into each hash value, so the support weight W_i^s can be effective and benefit the classification accuracy. Therefore, the number of hash bits should not be very large either. In fact, using 20 hash bits can result in one million different hash values, sufficiently representing half million cells in our dataset. In addition, using a large number of hash bits (e.g., 64 bits) may reduce the computational and memory efficiency, since the hash table is no longer an option owing to the memory constraint. Therefore, we have chosen 12 bits for this task, mapping half million cells to 4096 hash values and hence ensuring sound accuracy of classification without sacrificing the computational and memory efficiency. This is also demonstrated by our experiments shown in Fig. 6.4. Note that our model is able to generate accurate results within a certain range of parameter values, i.e., not that sensitive to parameters, making it suitable for the large-scale analysis. Furthermore, Fig. 6.4 also shows that our content-aware weighting scheme consistently improves the hashing method for classification accuracy, when using different number of hash bits.

Currently, we have validated our framework on around one thousand images with half million cells. We expect to apply it on much larger databases (e.g., hundreds of millions cells) or whole slide images in the future. In this case, parallel computing may be necessary to ensure the computational efficiency. Our framework for cell-level analysis can be straightforwardly parallelled. For example, the whole slide image can be divided as multiple patches, and each patch can be processed by one node of the cluster for cell segmentation and classification independently. Note that if holistic features are used, e.g., architecture features, such parallel computing can only be

applied on the cell detection and segmentation, but not the feature extraction, which needs to analyze the whole image simultaneously. In general, the computational efficiency of our framework is very promising and has the potential to handle large-scale databases.

6.4 Summary

In this introduce, we introduce a weighted hashing method to index and retrieve large-scale image databases, and employ this method to analyze histopathological images at cell-level. This weighting scheme alleviates the intrinsic problems of traditional hashing methods. It significantly improves the diagnosis accuracy of a challenging clinical problem, i.e., differentiating two types of lung cancers as the adenocarcinoma and squamous carcinoma using histopathological images. We envision that this large-scale image retrieval framework can provide useable tools to assist clinicians' diagnoses of cellular images and support efficient data management. Note that although this weighting scheme is specifically designed for cell-level analysis of histopathological images, resulting promising performance in this challenging application, it may also benefit the classification accuracy of other applications such as natural image categorization.

CHAPTER 7: FUSING MULTIPLE INDEXED FEATURES FOR RERANKING

7.1 Motivation

In previous chapters, we discussed using hashing based method to index and retrieve large-scale image databases. In order to boost the performance further, we need to fuse multiple types of information.

Generally, fusion can be carried out on the feature or rank-levels. In our context (i.e., differentiation of cancers), this means to combine different types of features in a histogram [131, 32] for learning-based classification, or to fuse the ordered results from CBIR methods [28, 132] and then classify via majority voting, both of which are fundamental problems. Unfortunately, many existing fusion methods still have limitations, especially in terms of the robustness, scalability, and generality. For example, feature-level fusion usually concatenates multiple feature vectors (e.g., the histogram of color features or texture features) and produces a new feature vector that has a higher dimensionality. However, when these features are heterogeneous (e.g., having significantly different dimensions and characteristics such as low-dimensional architecture feature [3] and high-dimensional appearance feature [137] in histopathological image analysis), feature-level fusion may not be able to effectively integrate their strengths. On the other hand, rank-level fusion combines different retrieval results (i.e., a list of retrieved images), obtained from using different types of features. This approach usually needs to decide which features should have an important role in the retrieval, which is quite difficult to determine online for a specific input with a large database.

In this chapter, we focus on the rank-level fusion of local and holistic features. Particularly, we use content-based image retrieval to discover relevant instances from

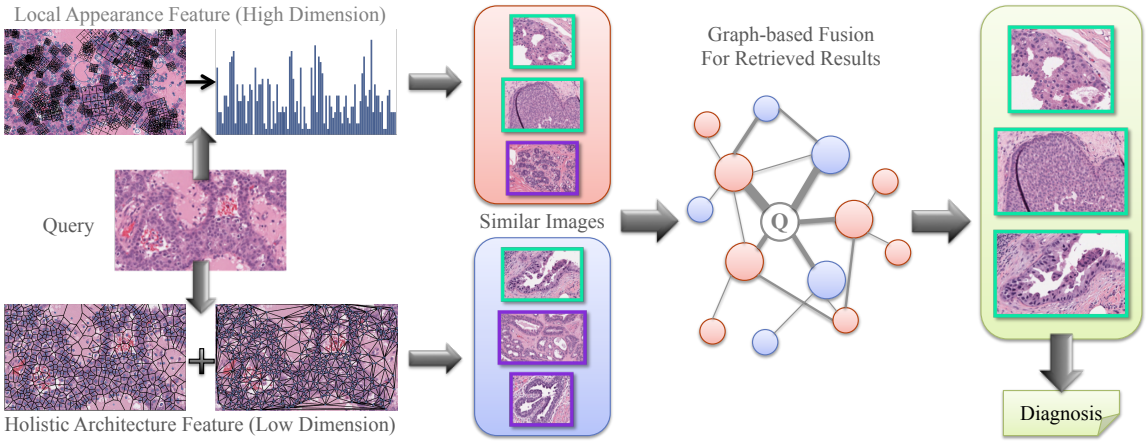


Figure 7.1: Overview of the graph-based feature fusion for image retrieval [136]. Both holistic architecture feature and local appearance feature are extracted and employed for image retrieval. The retrieval results are fused via the graph-based framework to improve the accuracy. Note that majority voting does not work in this example, since two ranks have no intersection.

an image database, which can be used to infer and classify the new data. Given image ranks (i.e., retrieval results) obtained from different features, a data-driven and graph-based method is employed for accurate, robust and efficient fusion, by evaluating the quality of each rank online [132].

7.2 Methodology

7.2.1 Overview

Fig. 7.1 shows the overview of our framework. From detected cells, we extract both holistic architecture features [3] and high-dimensional local appearance features [137] (i.e., 10,000 dimensions), both of which are used for image retrieval. To ensure the computational efficiency and scalability, the high-dimensional feature is compressed as tens of hash bits [137, 66]. Combining these complementary features is an intuitive approach to improve the accuracy. However, directly combining them at the feature-level may not be effective due to dramatically different representations. An alternative is to fuse them at the rank-level, i.e., retrieved images. The critical issue is how to measure and compare the quality of ranks on the fly, since fusion process should favor the rank with higher quality. As the similarity scores of retrieved results may vary

largely among queries and are not comparable between different ranks, a reasonable approach is to measure the *consistency* among the top candidates. Therefore, for each query image, we construct a weighted undirected graph from the retrieval results of one rank, where the retrieval quality or the relevance is modeled by the weights on the edges. These weights are determined by the overlap ratio (i.e., Jaccard similarity coefficient) of two neighborhood image sets. Then we fuse multiple graphs to one and perform a localized PageRank algorithm [81] to rerank the retrieval results according to their probability distribution. As a result, the fused retrieval results tend to be consistent among different feature representations.

7.2.2 Fusion of Heterogeneous Features

Based on the architecture feature and local appearance feature, k-nearest neighbors (kNN) algorithm can be naturally used to find similar cases of the input image. Since each feature can generate one set of results, i.e., a rank list, we conduct rank-level fusion for such heterogeneous features. This procedure includes graph construction, graph consolidation, and sub-graph selection, as shown in Fig. 7.2.

7.2.2.1 Graph Construction

Given a list of ranked results (i.e., retrieved images) by one type of features, such as the architecture or appearance feature, we assume that the consensus degree among the top candidates reveals the retrieval quality. Therefore, we first build a weighted graph using the constraints derived from the consensus degree, i.e., shared kNN. Setting the query as the graph centroid, we use its kNN as the first layer of nodes in the graph, and kNN of kNN as the second layer. Note that this setting is different from traditional methods using reciprocal kNN [86, 132], since such information is usually not available for medical image analysis, i.e., query is not included in the database. Neighboring nodes are connected by edges, whose weight can be defined as the ratio of their common neighbors, i.e., Jaccard similarity, which reflects the

confidence of including the connected nodes into the retrieval results. The weight between node i and i' is defined as:

$$w(i,i') = J(i,i') = \frac{|N_k(i) \cap N_k(i')|}{|N_k(i) \cup N_k(i')|} \tag{7.1}$$

where $|\cdot|$ denotes the cardinality, $N_k(i)$ and $N_k(i')$ include the images that are the top-k retrieved candidates using i and i' as the query, respectively. The range of edge weights is from 0 to 1, with $J(i,i') = 1$ implying that these two histopathological images share exactly the same set of neighbors, in which case we assume that they are highly likely to be similar.

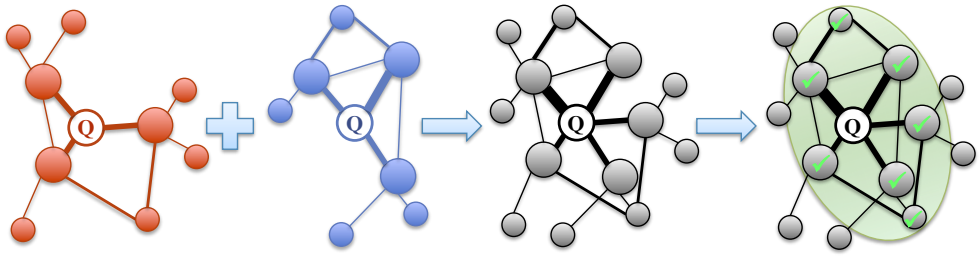


Figure 7.2: Procedures of our graph fusion, including graph construction (from two ranks, represented as blue and red graphs), graph consolidation (purple to represent nodes appearing in both graphs) and sub-graph selection.

7.2.2.2 Graph Consolidation

Multiple graphs, denoted as $G^m = (V^m, E^m, w^m)$, are constructed from the retrieved results of holistic and local features. They can be fused together in a natural way, by appending new nodes or consolidating edge weights of existing nodes in the resulting graph:

$$\begin{aligned} G = (V, E, w), \text{with } V = \cup_m V^m, E = \cup_m E^m, \\ \text{and } w(i,i') = \sum_m w^m(i,i') \end{aligned} \tag{7.2}$$

where $w^m(i,i') = 0$ for $(i,i') \notin E^m$. The rational of this fusion process is that though the rank lists or the similarity scores in different methods or features are not directly comparable, their Jaccard coefficients are comparable as they reflect the consistency of two nearest neighborhoods. In other words, this measure of consensus degree does not rely on the similarity scores, so it can be used and compared for different retrieval results from holistic and local features, ensuring the generality.

7.2.2.3 Sub-Graph Selection

After the candidates from both holistic and local features are fused via the graph consolidation, we need to rank them as per the relevance and select the most similar ones. This can be achieved by conducting a link analysis on the resulting graph, which is treated as a network. This is therefore equivalent to the PageRank problem [81] that discovers the probabilities of the nodes to be visited. Since this network is built by considering the retrieval relevance, naturally a node is more important or relevant if it has a higher probability to be visited. To compute the equilibrium state of the graph, we define the $|V| \times |V|$ transition matrix $\mathbf{P}$ as $P_{ii'} = w(i,i')/\deg(i)$ for $(i,i') \in E$, and 0 otherwise, where $\deg(i)$ means the degree or the number of neighbors for a specific node i. This matrix is row-stochastic, and the summation of each row equals to one. In the *intelligent surfer model* [87], a "surfer" probabilistically moves along the edges of G to different nodes, based on the transition matrix $\mathbf{P}$. We denote p_i^t as the probability for the surfer to be at node i at a time t and $p^t = (p_i^t)$. The equilibrium state of p is obtained by the query-dependent PageRank vector as a stationary point using the power method, indicating the relevance or similarity to the query image.

Once p has converged, the histopathological images are ranked according to their probabilities in p, where a higher probability reflects a higher relevance to the query in this equilibrium state of the graph. Using fused results, i.e., a new list of histopathological images from both features, majority voting can be employed for cancer differ-

entiation. To summarize, fusing heterogeneous features via graphs can significantly improve the performance of each individual feature, without sacrificing the scalability and generality.

7.3 EXPERIMENTS

7.3.1 Experimental Setting

Histopathological images of breast-tissue for this study were collected from the IU Health Pathology Lab (IUHPL) according to the protocol approved by the Institutional Review Board (IRB) [24]. All the slides were imaged using a ScanScope digitizer (Aperio, Vista, CA) available in the tissue archival service at IUHPL. 120 images (around 2250K pixels for each image) were gathered from 40 patients, 3 images per patient. 20 of these patients were labeled as benign and others are actionable, based on the majority diagnosis of nine board-certified pathologists. Leave-one-patient-out validation is used to evaluate the accuracy of classification. All parameters are tuned using cross validation to optimize the final result. The experiments were conducted on a 3.40 GHz CPU with 4 cores and 16G RAM, in a MATLAB implementation.

7.3.2 Evaluation of Individual Features

We employ two types of features, holistic and local, as the baseline methods for fusion. For holistic feature [3], the Voronoi diagram, Delaunay triangulation, minimum spanning tree are constructed and the nuclear density features are computed to model “architecture" of breast tissue, resulting in a 48-dimensional feature vector for each image. For local feature, 1500 to 2000 SIFT descriptors [70] are extracted from each image by detecting key points to describe the cell appearance. These descriptors are quantized into sets of cluster centers using bag-of-words [97], in which the feature dimension equals the number of clusters. Specifically, we quantize them into high-dimensional feature vectors with length 10,000. For efficiency and scalability we compress the high dimensional feature into 48 binary bits with kernelized

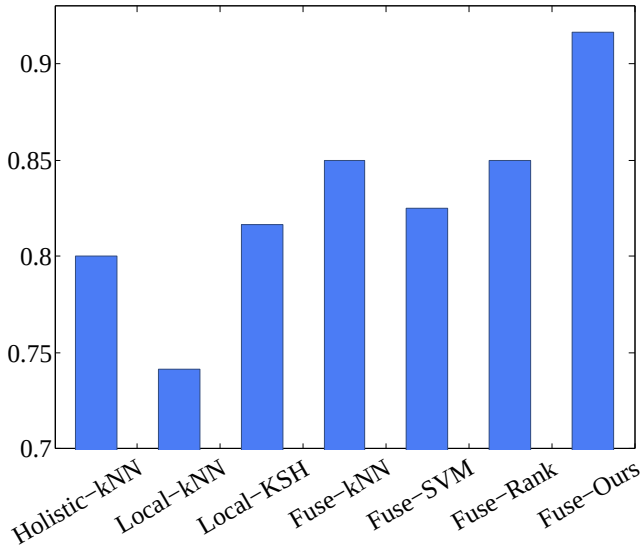


Figure 7.3: Quantitative comparison of the classification accuracy. We compare the performance of each single feature, and the fusion of both holistic and local features.

supervised hashing (KSH) algorithm [66]. Note that this binary representation is not compatible with the holistic feature. We first evaluate the performance of image retrieval using single feature such as the holistic feature, high-dimensional local feature and compressed binary feature. kNN and Support Vector Machine (SVM) are used as the baselines that have been widely employed for histopathological image analysis [3, 101, 92].

As shown in Fig. 7.3, both holistic and local features are able to generate reasonable results, i.e., around 80% accuracy. The only exception is that kNN fails in handling high-dimensional local feature, achieving only 74.17% accuracy. After compression with KSH, the binary codes improve the accuracy to 81.67%. In addition, using hashing representation also significantly improves the computational efficiency, i.e., thousands times faster than using original high-dimensional features, ensuring the scalability. Since both features are fairly effective but not perfect, and they should be complementary as they model different scales of information, it is natural to combine them for higher accuracy.

7.3.3 Evaluation of Feature Fusion

We compare our fusion framework with several classical methods for fusion, including both feature and rank-level approaches. For feature-level fusion, we normalize and concatenate different features into a histogram [32] and classify them with either kNN or SVM. Since the dimensions of features are largely different, it is not likely to obtain reasonable results without doing normalization. Therefore, normalization ensures that each feature contributes “equally" to the concatenated one [71]. For rank-level fusion, we combine different retrieval results via rank aggregation [28] and classify the query image with majority voting. Rank aggregation has been employed to fuse image retrieval results from similar types of features [44].

As shown in Fig. 7.3, concatenation of feature vectors marginally improve the classification accuracy, i.e., around 1-3% better than the baseline, due to the dramatically different characteristics of heterogeneous features. On the other hand, rank aggregation also merely improves the accuracy by 3%, since there may be no intersection among the top candidates retrieved by the local and holistic features. Our graph fusion method determines online which features should play a major role in the retrieval, in an unsupervised scheme. As a result, our fusion of heterogeneous features significantly improves the accuracy by around 10%, i.e., achieving 91.67% overall accuracy on this challenging problem. In addition, since this fusion process is applied on the retrieved results, i.e., a small subset of the whole dataset, it is very efficient and only takes milliseconds, ensuring promising scalability. Fig. 7.4 shows some retrieval results using our framework.

7.3.4 Discussions

In this section, we discuss the parameters and implementation issues of our system. Our fusion method only has one important parameter, i.e., k for constructing the graphs. As shown in Fig. 7.5, the accuracy is related to this parameter. For example,

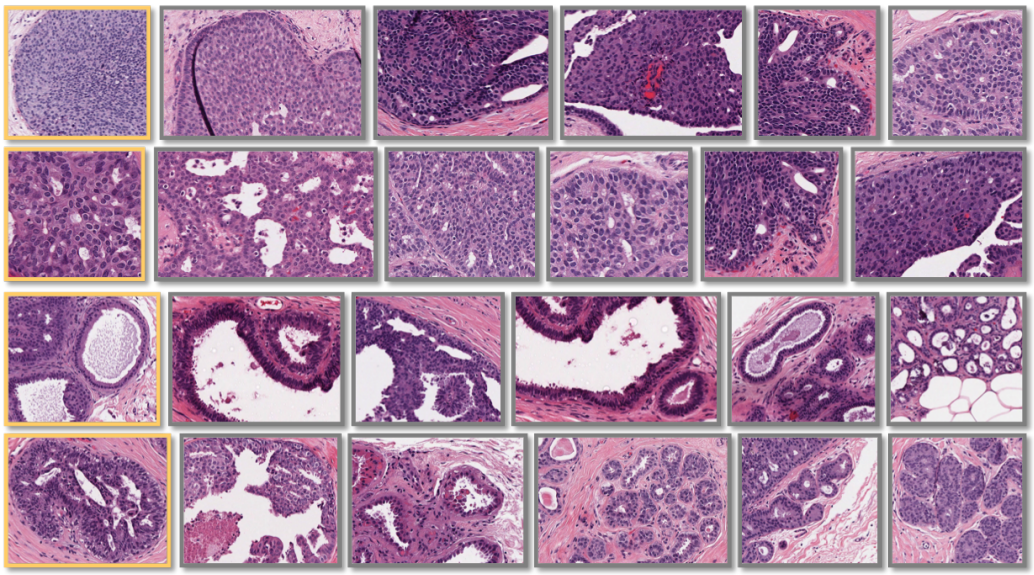


Figure 7.4: Retrieval results using our fusion framework. The first image in each row is the query, and the remaining ones are retrieval results. Top two rows are actionable cases, and bottom two rows are benign.

choosing a very small value for k (e.g., 3) indicates strong constraints of including nodes in the graph. Therefore, the resulting graphs usually do not have enough nodes. In other words, our graph fusion and reranking method cannot find enough candidates to select from. On the other hand, choosing a large value for k (e.g., 25) looses the constraints, so the graphs may have many nodes that are loosely related with the query. This also adversely affects the accuracy. Therefore, it is desired to choose a proper value of k. The motivation is to have sufficient and related candidates (i.e., nodes in graphs) incorporated into each graph, so our graph fusion algorithm can combine their strengths. Fig. 7.5 shows that our fusion results are consistently better than each of the baseline, In fact, it can achieve promising results (i.e., more than 90.0%) in a certain range of values, indicating that our method is not sensitive to small variations of k.

In our experiment, graph-based rank-level fusion significantly outperforms feature-level fusion and rank aggregation. However, this is not guaranteed and depends on the properties of features. When two types of features are heterogeneous, their histograms may have dramatically different properties, e.g., sparsity and dimensions. Our method

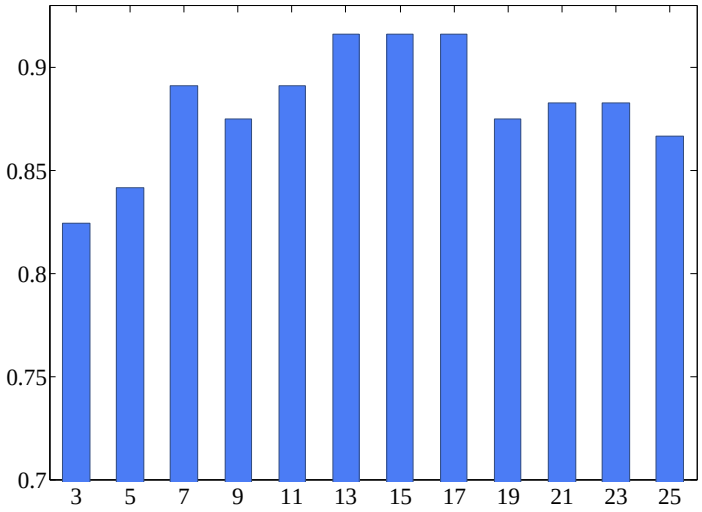


Figure 7.5: Evaluation of parameter k when constructing the graphs, ranging from 3 to 25.

becomes particularly useful, as it adaptively decides the quality of retrieval results on the fly. On the other hand, if these features have similar characteristics, e.g., features from multiple color spaces, they tend to generate similar ranks such that majority voting can be effective. In this case, rank aggregation or concatenation of histograms are able to achieve accurate results for fusion. We have conducted an experiment on fusing sub-types of architecture features. In fact, this 48-dimensional architecture feature is a concatenation of four holistic features, i.e., Voronoi features, Delaunay features, Minimum Spanning Tree features and Nucluear features, whose accuracy are 69.2%, 70.8%, 77.5% and 78.3%, respectively. Both feature-level fusion and our graph fusion achieves 80.0% accuracy, indicating that these four features are not heterogeneous.

7.4 Summary

In this chapter, we investigate the fusion of heterogeneous features for histopathological image analysis. Specifically, we employ a graph-based framework to fuse the holistic architecture feature and the local appearance feature that are generated from the cell detection results. These features are complementary but have dramatically different characteristics and representations, causing difficulties for traditional fusion

methods. Our framework can measure online the retrieval quality by the consistency of the neighborhoods of candidate images. Therefore, the fused results significantly improve the baseline using the single feature.

CHAPTER 8: CONCLUSIONS AND FUTURE DIRECTION

In this dissertation, we introduce solutions for large-scale and fine-grained image recognition. Particularly, we propose a series of methods in dealing with feature representation learning, indexing, and fusion, all of which are critical to the performance of this task. First, we learn feature representation through deep learning, which can be used to effectively differentiate fine-grained differences. Then, feature indexing via hashing or binary coding is utilized to enable real-time retrieval among large-scale databases. Finally, a rank-level feature fusion method is employed for more accurate retrieval results. We have validated our methods on synthetic images, natural images, and medical images. Regarding synthetic images, we have solid experiment results show that the feature we learned is invariant to the irrelevant factors. Regarding natural images, we have achieved state-of-the-art performance on four public fine-grained datasets. Regarding medical images, we have conducted the cell-level analysis of histopathological images, based on our large-scale feature indexing framework. The reason to investigate medical images is that they are ultra-fine-grained data and also have the significant impact, since it has many potential applications, including image-guided diagnosis, decision support, education in medical school, and efficient data management. For example, the efficient retrieval of relevant cases from medical databases will provide useable tools to assist clinicians' diagnoses and support efficient medical image data management, such as picture archiving and communication systems (PACS). More specifically, it provides efficient reasoning in large-scale medical image databases using techniques for scalable and accurate medical image retrieval in potentially massive databases to provide real-time querying for the most relevant and consistent instances (e.g., similar morphological profiles) for decision

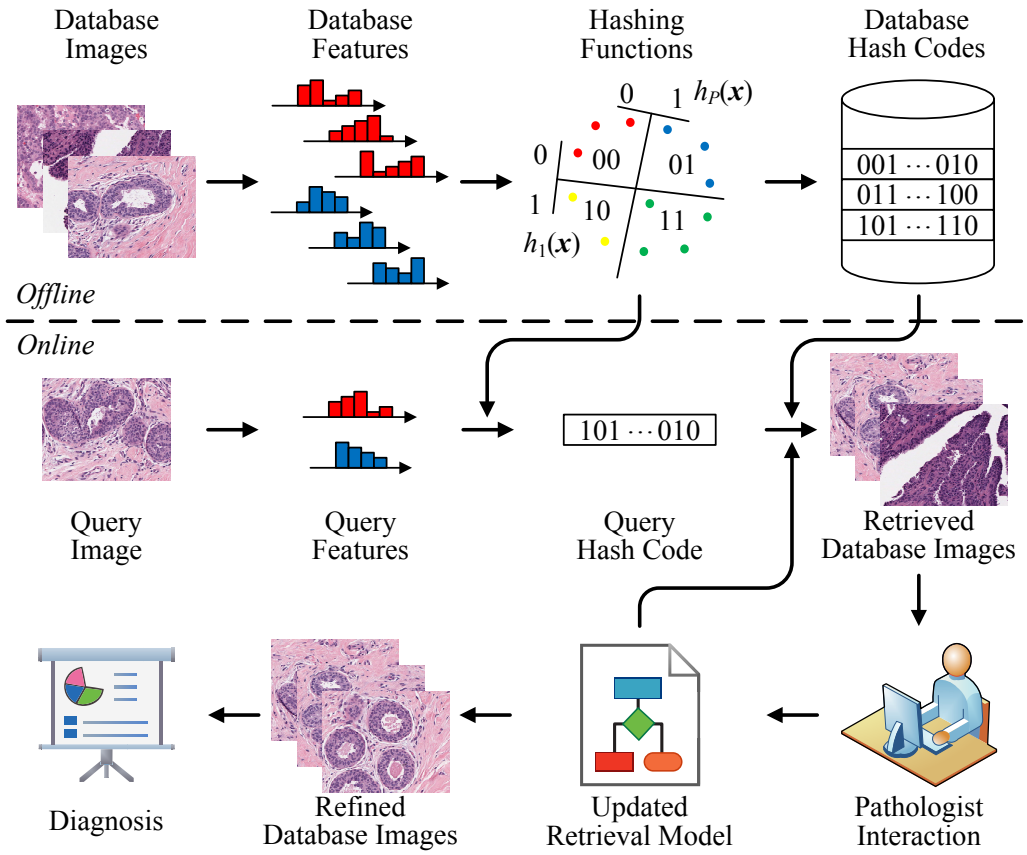


Figure 8.1: Example of incorporating domain knowledge from pathologists into the loop of hashing model updating.

support. In addition to the resulting tools for medical image processing, disease detection, and information retrieval, their use will allow for the exploration of structured image databases, in medical education and training.

In the future, we will focus on the intelligent interaction and visualization that integrates expert feedback and automated algorithms for efficient fine-grained image recognition [47, 115]. For example, in medical images, this system can support decision-making and provide a comprehensive understanding of the query results and supports semantic interaction functions. Interaction and visualization are another important yet challenging tool for effective computer aided diagnosis and medical data mining. To achieve our ultimate goal of assisting domain experts for efficient decision making and reasoning using fine-grained image databases, we plan to incorporate users in the loop to incorporate domain knowledge of experts. Fig. 8.1 shows

an example of interactively analyzing histopathological images. While the automated methods are designed to process millions of images, human users can only reasonably work with much fewer images at a time. The main challenge will be bridging the gap between the large-scale automated algorithms with the knowledge that domain experts can provide, but at much smaller scales. We plan to design a visual analysis system with a set of feature-based query, visualization, comparison, and learning methods for revealing the relevant image features and relationships. This system will support the analysis of the retrieved relevant image sets, extracted image features, and feature similarities among the retrieved image sets, and will provide efficient interaction methods to enhance the query algorithms and obtain finer-tuned results. To summarize, the components of large-scale retrieval and intelligent interaction will be coordinated for the purpose of scalable and interactive mining to provide a semantic interface between users and data through the language of feature similarities. The overall framework will be designed to address the challenges of both *scalable and interactive* mining using fine-grained image databases, and each aspect of the design and development will be driven by the goals of efficiency, robustness, and effective integration of user input.

REFERENCES

[1] H. C. Akakin and M. N. Gurcan. Content-based microscopic image retrieval system for multi-image queries. *IEEE Transactions on Information Technology in BioMedicine*, 16(4):758–769, 2012.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 819–826. IEEE, 2013.

[3] A. N. Basavanhally, S. Ganesan, S. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszewski, G. Bhanot, and A. Madabhushi. Computerized image-based detection and grading of lymphocytic infiltration in HER2+ breast cancer histopathology. *IEEE Transactions on Biomedical Engineering*, 57(3):642–653, 2010.

[4] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 955–962. IEEE, 2013.

[5] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2019–2026. IEEE, 2014.

[6] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, pages 663–676. Springer, 2010.

[7] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014.

[8] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.

[9] J. C. Caicedo, A. Cruz, and F. A. Gonzalez. Histopathology image classification using bag of features and kernel functions. In *Artificial Intelligence in Medicine*, pages 126–135. Springer, 2009.

[10] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *IEEE International Conference on Computer Vision*, pages 321–328. IEEE, 2013.

[11] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11:1109–1135, 2010.

[12] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep domain adaptation for describing people based on fine-grained clothing attributes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5315–5324, 2015.

[13] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546. IEEE, 2005.

[14] D. Comaniciu, P. Meer, and D. J. Foran. Image-guided decision support system for pathology. *Machine Vision and Applications*, 11(4):213–224, 1999.

[15] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[16] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Annual ACM Symposium on Computational Geometry*, pages 253–262. ACM, 2004.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[18] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 580–587. IEEE, 2013.

[19] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 2015.

[20] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In T. Jebara and E. P. Xing, editors, *International Conference on Machine Learning*, pages 647–655, 2014.

[21] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski. Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features. In *IEEE International Symposium on Biomedical Imaging*, pages 496–499, 2008.

[22] S. Doyle, M. Feldman, J. Tomaszewski, and A. Madabhushi. A boosted bayesian multiresolution classifier for prostate cancer detection from digitized needle biopsies. *IEEE Transactions on Biomedical Engineering*, 59(5):1205–1218, 2012.

[23] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3474–3481. IEEE, 2012.

[24] M. M. Dundar, S. Badve, G. Bilgin, V. Raykar, R. Jain, O. Sertel, and M. N. Gurcan. Computerized classification of intraductal breast lesions using histopathological images. *IEEE Transactions on Biomedical Engineering*, 58(7):1977–1984, 2011.

[25] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):373–378, 2003.

[26] I. El-Naqa, Y. Yang, N. P. Galatsanos, R. M. Nishikawa, and M. N. Wernick. A similarity learning approach to content-based image retrieval: application to digital mammography. *IEEE Transactions on Medical Imaging*, 23(10):1233–1244, 2004.

[27] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[28] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *ACM SIGMOD*, pages 301–312. ACM, 2003.

[29] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber. Using deep learning to enhance cancer diagnosis and classification. In *International Conference on Machine Learning*, 2013.

[30] D. J. Foran, L. Yang, et al. Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. *Journal of the American Medical Informatics Association*, 18(4):403–415, 2011.

[31] J. Fu, Y. Wu, T. Mei, J. Wang, H. Lu, and Y. Rui. Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging. In *IEEE International Conference on Computer Vision*, pages 1985–1993, 2015.

[32] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *IEEE International Conference on Computer Vision*, pages 221–228. IEEE, 2009.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[34] C. Goering, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2489–2496. IEEE, 2014.

[35] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. *Advances in Neural Information Processing Systems*, 2004.

[36] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.

[37] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 817–824. IEEE, 2011.

[38] H. Greenspan and A. T. Pinhas. Medical image categorization and retrieval for PACS using the GMM-KL framework. *IEEE Transactions on Information Technology in BioMedicine*, 11(2):190–202, 2007.

[39] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[40] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *IEEE International Conference on Computer Vision*, pages 309–316. IEEE, 2009.

[41] A. Hanbury, H. Müller, G. Langs, and B. H. Menze. Cloud-based evaluation framework for big data. In *FIA book 2013*, Springer LNCS, 2013.

[42] P.-W. Huang and Y.-H. Lai. Effective segmentation and classification for HCC biopsy images. *Pattern Recognition*, 43(4):1550–1563, 2010.

[43] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[44] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11, 2010.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[46] K. Jiang, Q. Que, and B. Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 4933–4941, 2015.

[47] T. Karaletsos, S. Belongie, C. Tech, and G. Rätsch. When crowds hold privileges: Bayesian unsupervised representation learning with oracle constraints. *Stat*, 1050:16, 2015.

[48] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[49] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011.

[50] J. Krause, J. Deng, M. Stark, and L. Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

[51] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015.

[52] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pages 554–561. IEEE, 2013.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[54] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, pages 1042–1050, 2009.

[55] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[56] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, 2012.

[57] A. Kumar, J. Kim, W. Cai, M. Fulham, and D. Feng. Content-based medical image retrieval: A survey of applications to multidimensional and multimodality data. *Journal of Digital Imaging*, 26(6):1025–1039, 2013.

[58] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[59] G. Langs, H. Müller, B. H. Menze, and A. Hanbury. VISCERAL: Towards large data in medical imaging - challenges and directions. In *MCBR-CDS MICCAI workshop*, volume 7723 of *Springer LNCS*, 2013.

[60] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. 2013.

[61] T. M. Lehmann, M. O. Güld, T. Deselaers, D. Keysers, H. Schubert, K. Spitzer, H. Ney, and B. B. Wein. Automatic categorization of medical images for content-based retrieval and data mining. *Computerized Medical Imaging and Graphics*, 29(2):143–155, 2005.

[62] S. Liao and S. Z. Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *IEEE International Conference on Computer Vision*, pages 3685–3693, 2015.

[63] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015.

[64] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. *IEEE International Conference on Computer Vision*, 2015.

[65] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *European Conference on Computer Vision*, pages 466–480. Springer, 2014.

[66] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2074–2081, 2012.

[67] X. Liu, J. He, D. Liu, and B. Lang. Compact kernel hashing with multiple features. In *ACM international conference on Multimedia*, pages 881–884. ACM, 2012.

[68] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[69] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

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

[71] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *International Journal of Computer Vision*, 90(1):88–105, 2010.

[72] Y. Mu, J. Shen, and S. Yan. Weakly-supervised hashing in kernel space. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3344–3351. IEEE, 2010.

[73] H. Müller, A. Geissbühler, and P. Ruch. ImageCLEF 2004: Combining image and multi-lingual search for medical image retrieval. In *Multilingual Information Access for Text, Speech and Images*, pages 718–727. Springer, 2005.

[74] H. Müller and J. Kalpathy-Cramer. The ImageCLEF medical retrieval task at ICPR 2010–information fusion. In *IEEE International Conference on Pattern Recognition*, pages 3284–3287. IEEE, 2010.

[75] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, 2004.

[76] National Cancer Institute. The cancer genome atlas retrieved from https://tcga-data.nci.nih.gov, 2013.

[77] K. Nguyen, A. K. Jain, and R. L. Allen. Automated gland segmentation and classification for gleason grading of prostate tissue images. In *IEEE International Conference on Pattern Recognition*, pages 1497–1500. IEEE, 2010.

[78] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[79] M. Norouzi and D. M. Blei. Minimal loss hashing for compact binary codes. In *International Conference on Machine Learning*, pages 353–360, 2011.

[80] M. Norouzi, D. M. Blei, and R. R. Salakhutdinov. Hamming distance metric learning. In *Advances in Neural Information Processing Systems*, pages 1061–1069, 2012.

[81] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[82] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012.

[83] O. M. Parkhi, A. Vedaldi, A. Zisserman, A. Vedaldi, K. Lenc, M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, K. Lenc, et al. Deep face recognition. *British Machine Vision Conference*, 2015.

[84] X. Qi, F. Xing, D. Foran, and L. Yang. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. *IEEE Transactions on Biomedical Engineering*, 59(3):754 –765, mar. 2012.

[85] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3716–3724, 2015.

[86] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Computer Vision and Pattern Recognition*, pages 777–784. IEEE, 2011.

[87] M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *Advances in Neural Information Processing Systems*, pages 1441–1448, 2001.

[88] A. J. Schaumberg, M. A. Rubin, and T. J. Fuchs. H&e-stained whole slide deep learning predicts spop mutation state in prostate cancer. *bioRxiv*, page 064279, 2016.

[89] F. Schnorrenberg, C. Pattichis, C. Schizas, and K. Kyriacou. Content-based retrieval of breast cancer biopsy slides. *Technology and Health Care*, 8(5):291–297, 2000.

[90] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

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

[92] O. Sertel, J. Kong, U. V. Catalyurek, G. Lozanski, J. H. Saltz, and M. N. Gurcan. Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading. *Journal of Signal Processing Systems*, 55(1-3):169–183, 2009.

[93] G. Sharma and B. Schiele. Scalable nonlinear embeddings for semantic category-based image retrieval. *IEEE International Conference on Computer Vision*, 2015.

[94] F. Shen, C. Shen, W. Liu, and H. Tao Shen. Supervised discrete hashing. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 37–45, 2015.

[95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003.

[97] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, 2003.

[98] Y. Song, W. Cai, and D. Feng. Hierarchical spatial matching for medical image retrieval. In *ACM International Workshop on Medical Multimedia Analysis and Retrieval*, pages 1–6. ACM, 2011.

[99] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[100] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[101] A. Tabesh, M. Teverovskiy, H.-Y. Pang, V. P. Kumar, D. Verbel, A. Kotsianti, and O. Saidi. Multifeature prostate cancer diagnosis and gleason grading of histological images. *IEEE Transactions on Medical Imaging*, 26(10):1366–1378, 2007.

[102] O. Tuzel, L. Yang, P. Meer, and D. J. Foran. Classification of hematologic malignancies using texton signatures. *Pattern Analysis and Applications*, 10(4):277–290, 2007.

[103] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, et al. Understanding objects in detail with fine-grained attributes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3622–3629. IEEE, 2014.

[104] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[105] C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, and S. Belongie. Similarity comparisons for interactive fine-grained categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 859–866. IEEE, 2014.

[106] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[107] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang. Multiple granularity descriptors for fine-grained categorization. In *IEEE International Conference on Computer Vision*, pages 2399–2406, 2015.

[108] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3424–3431. IEEE, 2010.

[109] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2393–2406, 2012.

[110] J. Wang, W. Liu, S. Kumar, and S.-F. Chang. Learning to hash for indexing big data-a survey. *Proceedings of the IEEE*, 104(1):34–57, 2016.

[111] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1386–1393. IEEE, 2014.

[112] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *IEEE International Conference on Computer Vision*, pages 17–24. IEEE, 2013.

[113] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[114] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2008.

[115] M. J. Wilber, I. S. Kwak, D. Kriegman, and S. Belongie. Learning concept embeddings with combined human-machine expertise. In *IEEE International Conference on Computer Vision*, pages 981–989. IEEE, 2015.

[116] L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[117] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[118] S. Xie, T. Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 580, 2015.

[119] F. Xing, H. Su, J. Neltner, and L. Yang. Automatic ki-67 counting using robust cell detection and online dictionary learning. *IEEE Transactions on Biomedical Engineering*, 61(3):859–870, March 2014.

[120] F. Xing and L. Yang. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review. *IEEE reviews in biomedical engineering*, 2016.

[121] Z. Xu, S. Huang, Y. Zhang, and D. Tao. Augmenting strong supervision using web data for fine-grained categorization. In *IEEE International Conference on Computer Vision*, pages 2524–2532, 2015.

[122] L. Yang, W. Chen, P. Meer, G. Salaru, L. A. Goodell, V. Berstis, and D. J. Foran. Virtual microscopy and grid-enabled decision support for large-scale analysis of imaged pathology specimens. *IEEE Transactions on Information Technology in BioMedicine*, 13(4):636–644, 2009.

[123] L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. Hoi, and M. Satyanarayanan. A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):30–44, 2010.

[124] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3973–3981. IEEE, 2015.

[125] S. Yang, L. Bo, J. Wang, and L. G. Shapiro. Unsupervised template learning for fine-grained object recognition. In *Advances in Neural Information Processing Systems*, pages 3122–3130, 2012.

[126] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2249–2256. IEEE, 2010.

[127] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[128] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 192–199. IEEE, 2014.

[129] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[130] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *IEEE International Conference on Computer Vision*, pages 729–736. IEEE, 2013.

[131] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas. Automatic image annotation using group sparsity. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3312–3319, 2010.

[132] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas. Query specific rank fusion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):803–815, April 2015.

[133] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *European Conference on Computer Vision*, pages 660–673. Springer, 2012.

[134] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific rank fusion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):803–815, 2015.

[135] X. Zhang, H. Dou, T. Ju, J. Xu, and S. Zhang. Fusing heterogeneous features from stacked sparse autoencoder for histopathological image analysis. *IEEE journal of biomedical and health informatics*, 20(5):1377–1383, 2016.

[136] X. Zhang, H. Dou, T. Ju, and S. Zhang. Fusing heterogeneous features for the image-guided diagnosis of intraductal breast lesions. In *IEEE International Symposium on Biomedical Imaging*, pages 1288–1291. IEEE, 2015.

[137] X. Zhang, W. Liu, M. Dundar, S. Badve, and S. Zhang. Towards large-scale histopathological image analysis: Hashing-based image retrieval. *IEEE Transactions on Medical Imaging*, 34(2):496–506, Feb 2015.

[138] X. Zhang, W. Liu, and S. Zhang. Mining histopathological images via hashing-based scalable image retrieval. In *IEEE International Symposium on Biomedical Imaging*. IEEE, 2014.

[139] X. Zhang, H. Su, L. Yang, and S. Zhang. Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5361–5368, 2015.

[140] X. Zhang, H. Su, L. Yang, and S. Zhang. Weighted hashing with multiple cues for cell-level analysis of histopathological images. In *International Conference on Information Processing in Medical Imaging*, pages 303–314. Springer, 2015.

[141] X. Zhang, F. Xing, H. Su, L. Yang, and S. Zhang. High-throughput histopathological image analysis via robust cell segmentation and hashing. *Medical Image Analysis*, 26(1):306–315, 2015.

[142] X. Zhang, L. Yang, W. Liu, H. Su, and S. Zhang. Mining histopathological images via composite hashing and online learning. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 479–486. Springer, 2014.

[143] X. Zhang, F. Zhou, Y. Lin, and S. Zhang. Embedding label structures for fine-grained feature representation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[144] L. Zheng, A. W. Wetzel, J. Gilbertson, and M. J. Becich. Design and analysis of a content-based pathology image retrieval system. *IEEE Transactions on Information Technology in BioMedicine*, 7(4):249–255, 2003.

[145] X. S. Zhou, S. Zillner, M. Moeller, M. Sintek, Y. Zhan, A. Krishnan, and A. Gupta. Semantics and CBIR: a medical imaging perspective. In *ACM International Conference on Content-Based Image and Video Retrieval*, pages 571–580. ACM, 2008.