ABSTRACT

Title of dissertation: LEVERAGING MULTIPLE FEATURES FOR IMAGE RETRIEVAL AND MATCHING

Fan Yang, Doctor of Philosophy, 2016

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

The goal of image retrieval and matching is to find and locate object instances in images from a large-scale image database. While visual features are abundant, how to combine them to improve performance by individual features remains a challenging task. In this work, we focus on leveraging multiple features for accurate and efficient image retrieval and matching.

We first propose two graph-based approaches to rerank initially retrieved images for generic image retrieval. In the graph, vertices are images while edges are similarities between image pairs. Our first approach employs a mixture Markov model based on a random walk model on multiple graphs to fuse graphs. We introduce a probabilistic model to compute the importance of each feature for graph fusion under a naive Bayesian formulation, which requires statistics of similarities from a manually labeled dataset containing irrelevant images. To reduce human labeling, we further propose a fully unsupervised reranking algorithm based on a submodular objective function that can be efficiently optimized by greedy algorithm. By maximizing an information gain term over the graph, our submodular function

favors a subset of database images that are similar to query images and resemble each other. The function also exploits the rank relationships of images from multiple ranked lists obtained by different features.

We then study a more well-defined application, person re-identification, where the database contains labeled images of human bodies captured by multiple cameras. Re-identifications from multiple cameras are regarded as related tasks to exploit shared information. We apply a novel multi-task learning algorithm using both low level features and attributes. A low rank attribute embedding is joint learned within the multi-task learning formulation to embed original binary attributes to a continuous attribute space, where incorrect and incomplete attributes are rectified and recovered.

To locate objects in images, we design an object detector based on object proposals and deep convolutional neural networks (CNN) in view of the emergence of deep networks. We improve a Fast RCNN framework and investigate two new strategies to detect objects accurately and efficiently: scale-dependent pooling (SDP) and cascaded rejection classifiers (CRC). The SDP improves detection accuracy by exploiting appropriate convolutional features depending on the scale of input object proposals. The CRC effectively utilizes convolutional features and greatly eliminates negative proposals in a cascaded manner, while maintaining a high recall for true objects. The two strategies together improve the detection accuracy and reduce the computational cost.

Leveraging Multiple Features for Image Retrieval and Matching

by

Fan Yang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Hector Corrada Bravo
Professor Rama Chellappa
Professor Ramani Duraiswami
Professor Dana Nau

© Copyright by
Fan Yang
2016

Dedication

To my dearest wife and parents.

Acknowledgments

Firstly, I would like to thank my advisor, Prof. Larry Davis. Over the past years, he has always been kind, knowledgeable, approachable and supportive. He gave me the opportunity to work on different exciting and interesting projects, while allowing me to explore various topics of my own interest. I would never have made such achievement without his invaluable guidance, as well as substantial support, both academically and financially. Secondly, I would like to thank Prof. Hector Corrada Bravo, Prof. Rama Chellappa, Prof. Ramani Duraiswami and Prof. Dana Nau for sparing time to serve as my dissertation committee, as well as providing helpful feedbacks and comments during the dissertation preparation.

I would like to thank my collaborator, Dr. Zhuolin Jiang, for his generous help during my early Ph.D. years. Thanks to my internship mentors, Dr. Bogdan Matei and Dr. Mayank Bansal from SRI International, and Dr. Wongun Choi and Dr. Yuanqing Lin from NEC Laboratories America Inc. Working with them is a pleasure and results in fruitful research outcome.

I also would like to thank my lab-mates who have closely worked with me: Joe Ng, Chi Su, Vlad Morariu and Guangxiao Zhang. I learned a lot from them and was constantly enlightened by their inspiring ideas on research. Thank all my friends at the University of Maryland and other places: Yaming Wang, Shuyang Su, Chengxi Ye, Hua He, Ang Li, Yangmuzi Zhang, Bo Tian, Xi Chen, Xiyang Dai, *etc.* The time we spent together has been a colorful and enjoyable part of my memory.

Thank my parents for giving birth to me and raising me up. They are the best

parents and educators. For years I have been far away from them and could not stay by their side. I owe them a lot. Finally, deepest thanks to my wife and soul mate, Jing Pan, for unconditionally supporting me and sacrificing for the family. We had hard times, but will have a bright future together full of happiness as I believe and promise.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In this work, we research on two fundamental aspects of computer vision: searching for similar images from a database, and detecting and recognizing objects from images. The two aspects are different in that the former focuses on comparing and inferring the similarity of images, while the latter learns to precisely locate objects in images. Nevertheless, both involve analyzing and understanding the content of images for decision making, and are closely related in a practical vision system. Formally, the process that using images as input without textual information to search for similar images is referred to as content-based image retrieval. As for object detection, it has been extensively studied and there are abundant research works. Here we limit our focus to the approaches based on deep neural networks that have gained great popularity recently and shown excellent performance on various vision tasks.

1.0.1 Content-based Image Retrieval

Content-based image retrieval has been studied for decades due to its importance in practical applications, such as commercial search engines, marketing and branding, and near-duplicate removal. A content-based image retrieval system gen-

erally works as follows. First, visual features are extracted from database and query images as image representation. Second, for database images, feature vectors are stored and usually indexed in an optimized way to describe the structure of database for efficient retrieval. Finally, visually similar images are discovered and ranked by calculating the distances between the feature vectors of query images and database images. Database images with smaller distance to the query image are deemed as more similar and thus ranked higher.

Regarding features, most of existing approaches adopt a single feature such as bag-of-words (BoW) [1,3,4,8–10], Fisher vectors (FV) [11,12], vector locally aggregated descriptors (VLAD) [13], or their improved versions [14–16]. However, these methods heavily relies on keypoint detectors, thus are not robust enough against blurred images due to camera motion and objects that occupies only a small portion of the entire image. In these cases, only a limited number of or even no keypoints can be extracted, which makes the keypoint-based approaches vulnerable. On the other hand, global features, such as color histograms, are more powerful to capture higher level information compared to local features, which may help us locate the correct object accurately and retrieve them effectively. Nevertheless, global features ignore the subtle details of objects, which are crucial in image retrieval to accurately discriminate different objects. Therefore, a single feature may not effectively handle all the different variations and thus combining multiple complementary features is a way to exploit the information that cannot be found by a single feature alone.

Generally, there are two ways of combining multiple features: early fusion and late fusion. In early fusion, weights for multiple feature vectors are learned

from training data and used to concatenate raw feature vectors. In contrast, only pairwise similarities between images with respect to multiple features are taken into consideration for combining initial results from individual features. Since it is more flexible and feature-agnostic, we choose late fusion for multi-feature fusion, and propose several approaches for various image search scenarios, including generic image retrieval and a more well-defined problem, person re-identification.

1.0.2 Object Detection with Deep Neural Networks

Object detection has been extensively studied due to its importance in image analysis and understanding. Before the emergence of deep neural networks, the deformable part model (DPM) [17] with hand-crafted features, such as histogram of gradients (HoG), has been the state-of-the-art object detector for decades. With the extraordinary representative and discriminative capability of deep neural networks, the effort on designing features and choosing appropriate learning algorithms to obtain an effective object detector switches to tunning network architectures and learning parameters. In this way, one can easily obtain a powerful object detector that allows end-to-end detection without much human intervention.

Nevertheless, designing an effective and efficient object detector based on deep neural networks still remains a challenging problem. Although deep networks provides highly discriminative features, yet the computational cost still remains too large to detect objects for practical use. We address the problem of high computational cost and propose a new approach to accelerate detection, apart from

improving the detection accuracy.

1.1 Outline of Thesis

In Chapter 2, we propose a simple yet effective framework for multi-feature fusion based on graphical models for generic image retrieval, which requires similarities from annotated similar/dissimilar image pairs from a training database. This chapter is based on our work in [18]. For each feature, given the query and initially retrieved images, we construct an undirected graph whose vertices represent these images and in which edge strength is the pairwise similarity score between images. We employ a mixture Markov model, which is based on a random walk model on multiple graphs, to fuse multiple graphs into one. We introduce a probabilistic model to compute the importance of each feature under a naive Bayesian formulation that depends only on the statistics of similarity scores inferred from the annotated similar/dissimilar image pairs. The probability of walking between graphs is determined by the probabilistic model that measures the probability of a given similarity from similar images or dissimilar images.

In Chapter 3, we present a fully unsupervised approach without the requirement of annotated similar/dissimilar pairs of images, which is based on our work in [19]. In this approach, we construct a submodular objective function that consists of two terms: an information gain term and a relative ranking consistency term. To compute the information gain, we again represent each initial ranked list as an undirected graph, where the structure is then modeled as a transition matrix

under the assumption of a random walk on a graph. We select a subset of retrieved images by maximizing the information gain over the graph, which maximizes the mutual information between the selected subset and unselected nodes in the graph. The relative ranking consistency term exploits inter-relationships among multiple ranked lists obtained by different features. If relative ranks between two images are consistent across multiple ranked lists, the ranking relationship between them is considered reliable and captured by the relative ranking consistency term. Additionally, the relative ranking consistency term encourages selecting images that are similar to the query but only found and highly ranked by a small number of features. The final submodular objective function combines both the relationships among retrieved images from a single feature and the relative ranks of image pairs across different features, thereby improving initial retrieval results obtained by multiple independent features.

In Chapter 4, we focus on a more well-defined problem, person re-identification, which can be considered as a special case of generic image retrieval. This chapter is based on our work in [20]. In person re-identification, the database usually contains a lot of well labeled data that allows more sophisticated learning algorithms. Apart from low level features, we incorporate high level semantics, referred to as attributes, into the framework in view of their discriminative power and consistency across different representation spaces. Specifically, we propose a novel multi-task learning framework with low rank attribute embedding for person re-identification. Re-identifications from multiple cameras are regarded as related tasks to exploit shared information to improve accuracy. Since attributes are generally correlated,

we introduce a low rank attribute embedding into the multi-task learning formulation to embed original binary attributes to a continuous attribute space, where incorrect and incomplete attributes are rectified and recovered to better describe people. Low level features and embedded attributes are concatenated to form the final feature vectors. The learning objective function consists of a quadratic loss regarding class labels and an attribute embedding error, which is solved by an alternating optimization procedure.

Going one step further from finding similar images from the database, we would like to localize objects in images and recognize their categories. In Chapter 5, we propose an object detector based on deep convolutional neural networks (CNN). This chapter is based on our work in [21]. Inspired by the recent progress in object detection with CNNs, we investigate two new strategies to detect objects accurately and efficiently using deep CNNs: 1) scale-dependent pooling and 2) layer-wise cascaded rejection classifiers. The scale-dependent pooling (SDP) improves detection accuracy by exploiting appropriate convolutional features depending on the scale of the candidate object proposal. The cascaded rejection classifiers (CRC) effectively utilize convolutional features and eliminate negative bounding boxes in a cascaded manner, which greatly speeds up the detection while maintaining high accuracy. In combination of the two, our method achieves significantly better accuracy compared to other state-of-the-arts in two challenging datasets, while being more efficient.

Chapter 6 provides the conclusion to the thesis.

Chapter 2: Image Retrieval by Mixture Markov Model and Diffusion

2.1 Introduction

The image retrieval task focuses on searching for same/similar images given a query image without textual information at the *instance-level*, which contains a particular object, rather than at the *category-level*, where we only need to differentiate from object categories, such as persons, animals and scenes. The bag-of-words (BOW) representation [8] based on local features, such as the SIFT descriptor [22], is widely used in retrieval systems. Numerous improvements with respect to performance and scalability of the original BOW representation have been proposed [1, 3, 4, 9, 10]. To reduce the dimensionality of the standard BOW representation that requires millions of visual words, Jégou *et al.* [13] introduced the vector of locally aggregate descriptors (VLAD) to achieve a trade-off between memory footprint and retrieval performance. Despite their power in capturing local patterns of an object, local features such as SIFT and VLAD descriptors may not be suitable for describing the global characteristics of an image, which are well captured by global features. We present an example showing that different types of objects have different appearance information, thus requiring different types of features. In Figure 2.1, we show several images from two logo categories, *pepsi* and *apple*. The

Figure 2.1: Samples of *pepsi* and *apple* logos. Note that the *pepsi* logos exhibit various scale and rotational changes but the color distribution is relatively constant. In contrast, the *apple* logos exhibit varied colors, but consistent shape.

apple logo is of various colors, while its shape is relatively consistent across different samples. Therefore, shape descriptors are more appropriate to describe *apple* logos than color features. In contrast, the *pepsi* logo has a distinct color distribution that is composed of blue, red and white, although it exhibits various scale and rotational changes. In this case, as a global feature, color is more powerful to capture higher level information compared to local features, which may help us locate the correct logos accurately and retrieve them effectively.

However, how to combine multiple features still remains an open question. Usually, to better capture distinctive local and global patterns of images from a large collection of images, the dimensionality of feature vectors has to be extremely high. One has to use millions of visual words for constructing BoW vectors or tens of thousands of dimensions for Fisher Vectors (FV) [23] to obtain good performance. It is prohibitively expensive both to store all feature vectors for a database containing millions of images, as well as to learn weights from those features using any classifiers. In addition, due to large variation of dimensionality among different features, it is even more challenging to determine the relative importance of individual features if they are simply concatenated, since the performance of the concatenated

feature is prone to be dominated by high dimensional features such as BoW vectors. Furthermore, for retrieval tasks, we can only obtain a limited amount of labeled samples because manual annotation for millions of images is impractical, while the appearance of database images can be quite diverse. Moreover, we do not have any prior information of and cannot make any assumption on the characteristics of queries, which might be very different from the database images. Learning on a small set of annotated samples, which do not sufficiently represent the characteristics of the entire database and queries, is likely to generalize poorly.

In this chapter, we present a retrieval and reranking approach that utilizes pairwise similarity scores between images using multiple features rather than directly combining raw feature vectors. By introducing additional supervised information, we are able to combine similarity scores effectively, which leads to more accurate retrieval results compared with existing methods.

2.2 Related Work

2.2.1 Image Retrieval by Single Feature

We first introduce and discuss several types of features widely used in classic image retrieval systems to provide a better context of our work. The BoW feature is usually adopted as an image representation. Similarities between BoW feature vectors of a query image and dataset images are then computed for retrieval [8]. With BoW representations, Sivic *et al.* [8] applied standard term frequency-inverse document frequency (tf-idf) method to image retrieval. A hierarchical clustering

algorithm [9] was then proposed to construct a vocabulary tree which reduces the computational cost and is scalable to large scale datasets. Contextual weighting [10] was further applied to vocabulary trees to increase the discriminative ability of visual words. Instead of quantizing a descriptor to a single visual word, assigning it to multiple words results in more discriminative BoW vectors and thus achieves better performance [2, 24]. To compensate for the spatial information loss in the standard BoW-based approach, spatial verification [1] was proposed to match SIFT descriptors between images at the cost of extra storage space and computation time. Query expansion [3–5] has been widely applied to rerank initially retrieved images, where a small portion of top ranked images serve as additional queries and are fed into the retrieval system again to further explore similar images. Bundling min-hash [25] was also proposed to group locally close keypoints and encode them using min-hash. A few works attempt to address the "burstiness problem", where a large amount of keypoints from repetitive patterns in the background dominate the image representation. A statistical model was learned in [26] to down-weight the scores of keypoints which are frequently matched in incorrect detections. Multiple match removal (MMR) [24] was also proposed, where each keypoint votes only once for an image in the database, so that repetitive matches from a few keypoints can be effectively removed. Some improvements such as Hamming embedding with geometric constraints [6], dataset-side feature augmentation [5] and co-occurrences of visual words [14] have achieved state-of-the-art results.

Focusing on feature design, Jégou *et al.* [13] proposed the vector locally aggregated descriptor (VLAD) as a compact representation. It achieved good results

while requiring less storage compared to the BoW feature. Improvements on VLAD have been presented, including PCA and whitening [14] and signed square root (SSR) on VLAD vectors [15]. Multi-VLAD [16] was later proposed to construct and match VLAD features of multiple levels from an image to improve localization accuracy. RootSIFT [5] was proposed to address the burstiness problem with standard BoW features by using the Hellinger kernel on the original SIFT. GIST descriptors and Fisher Vector (FV) have also been evaluated for large-scale image retrieval.

2.2.2 Image Retrieval by Multiple Features

Although a single feature can achieve good retrieval results, better performance is anticipated if retrieved results from multiple features are properly fused. This is because they usually describe images from complementary perspectives. Recent works on fusing multiple features for image retrieval have been proposed, such as multi-modal graph learning [27], query-specific graph fusion [28] and co-regularized multi-graph learning [29]. In [12], multiple attribute features are combined by averaging outputs of SVM classifiers. The score vector is then concatenated with Fisher Vectors after normalization and dimensionality reduction. Graph-based techniques are also widely used in the literature. Wang *et al.* [27] proposed a graph-based learning algorithm to infer weights of features. Weights of individual features are learned statistically from the retrieved results given a large set of queries, and thus this method is not flexible if we do not have any information of queries beforehand. Zhang *et al.* [28] converted initial ranked lists by individual features to undirected

graphs by calculating the k-reciprocal nearest neighbors of each image, so that the connectivity of vertices in each graph captures the relationships among database images. Similarities between images are evaluated by Jaccard similarity instead of original similarities from comparing distance between feature vectors. Multiple graphs are then equally summed up. However, Jaccard similarity is too coarse to describe the pairwise relationships of images as it only captures the graph structure rather than the degree of similarity. Deng *et al.* [29] imposed intra-graph and inter-graph constraints in a supervised learning framework which requires image attribute information. A complicated multi-graph learning algorithm with co-regularization was applied to learn a weight matrix from multiple graphs. Attributes serve as weak labels to learn the most representative images from graphs, which are called "anchor" images, to align multiple graphs. Similarly, Zhang *et al.* [30] also utilized attributes learned from a large dataset apart from the retrieval database. 1000 attributes are learned from ImageNet database, so that each image in the retrieval database can be represented as a 1000-dimensional feature vector. Nearest neighbors of each database image are obtained by comparing the attribute vectors. These nearest neighbors provide additional information to refine the inverted file that is originally constructed by SIFT visual words. Recently, Zheng *et al.* [31] constructed a 2D indexing file using SIFT and color visual words. Similar to computing SIFT descriptors, color features are extracted around each detected keypoints and clustered to form color dictionary. Different from the previous works using a 1D inverted indexing file, the 2D indexing file indexes two features jointly as a regular grid, while each feature occupies one dimension of the grid. Therefore, only images with same

visual words from both features can be matched. However, it is not clear how to deal with more than two features.

2.2.3 Multi-feature Learning

With respect to multi-feature learning, there are numerous feature fusion algorithms available, while we limit our focus only on some representative works closely related to our work. Multi-kernel learning (MKL) [32–34] was widely used to find the optimal combination of kernels for image classification, where each feature type can be mapped to different kernels. Partial Least Squares (PLS) analysis [35] was applied to dimension reduction of a high dimensional vector formed by multiple feature vectors, which implicitly selects the most important features. Canonical Correlation Analysis (CCA) [36] was also effective to learn relationships of two sets of features. A hierarchical regression algorithm was proposed in [37] to exploit the information from individual features, where the manifold structure of different feature spaces is preserved. For cartoon image retrieval, Yang *et al.* [38] proposed a bi-distance metric learning algorithm to learn a distance metric from heterogeneous features. Ye *et al.* [39] decomposed multiple score matrices by multiple features as a low rank matrix plus feature-specific sparse errors. Fernando *et al.* [40] proposed to learn logistic regression models with sparsity regularization to determine weights for visual words from multiple dictionaries for image classification.

2.3 Proposed Approach

2.3.1 Overview

We present a supervised, data-driven approach to fuse multiple features to rerank database images based on a graph representation. For each feature, given the query and initially retrieved images, we construct an undirected graph whose vertices represent these images and in which edge strength is the pairwise similarity score between images. We employ a mixture Markov model to combine multiple graphs into one. In contrast to [28], where graphs are equally weighted, we introduce a probabilistic model to compute the importance of each feature under a naive Bayesian formulation, which depends only on the statistics of image similarity scores. Despite its simplicity, the proposed probabilistic model consistently improves retrieval performance after reranking. Instead of reranking the retrieved images directly from the fused graph, we employ an iterative diffusion algorithm, which propagates similarity scores throughout the graph to alleviate the effect of noise. This further improves the retrieval performance. In particular, we apply the locally constrained diffusion process (LCDP) [41] on the localized K-NN graph to obtain the refined similarity scores.

2.3.2 Graph Construction

Given a query image, an initial retrieval algorithm is performed to rank images from a dataset according to the similarity scores between the query image and

dataset images. Suppose we have r features, each of which is a type of feature focusing on a specific aspect of an image. For each feature $\mathcal{M}_m$, the similarity between images $\mathcal{I}_i$ and $\mathcal{I}_j$, denoted as $s_{i,j}^m$, where $0 \leq s_{i,j}^m \leq 1$, is obtained by comparing two feature vectors. Generally, the initial rankings produced from different features will not agree; our hope is that by appropriately fusing them we will obtain an overall more accurate ranking.

From initial retrieval results of all r features, we obtain n unique images totally, including the initial query. The pairwise relationships with respect to feature $\mathcal{M}_m$ among these images is represented by a graph $G_m = (V_m, E_m, e_m)$ where vertices V_m are images connected by edges E_m with edge strength e_m. The e_m is the similarity between two images under feature $\mathcal{M}_m$. The original dataset may contain millions of images, resulting in a very long ranked list of retrieved images for each query and thus a huge graph. Therefore, based on an estimation or prior knowledge of the possible number of similar images in the database, we only choose the top L retrieved images for each feature to construct a tractable graph. The ranked list of top L retrieved images is referred as a short list. We denote the union of nodes from all graphs as V. For each graph G_m, we add vertices which are from V but not initially retrieved by feature $\mathcal{M}_m$ into the graph. Edges connecting a previously missing vertex and initially retrieved vertices in the graph are also added. In this way, we complete each graph with missing vertices, so that each graph has the same set of vertices V. Even if short lists from multiple features are disjoint, by completing graphs, we include pairwise relationships between vertices in these short lists and may still improve performance.

Each graph can be represented as a symmetric matrix $\mathbf{S}_m \in \mathbb{R}^{n \times n}$ with diagonal elements $s^m_{i,i} = 1$, known as an affinity matrix. Each element in the affinity matrix $\mathbf{S}_m$ represents the edge strength between nodes v_i and v_j in the graph. The i-th row in the affinity matrix $\mathbf{S}_m$ contains similarity scores between image $\mathcal{I}_i$ and all other images (including $\mathcal{I}_i$ itself). For r features, we have a set of r graphs $\mathcal{G} = \{G_1, G_2, ..., G_r\}$ corresponding to a set of affinity matrices $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_r\}$ of the same size. The similarity score $s^m_{i,j}$ between a query $\mathcal{I}_i$ and a dataset image $\mathcal{I}_j$ that was not retrieved by feature $\mathcal{M}_m$ is simply set to 0.

2.3.3 Multi-feature Graph Fusion

After obtaining affinity matrices in $\mathcal{S}$ from Section 2.3.2, our goal is to fuse graphs in $\mathcal{G}$ using these matrices. Affinity matrices should be complementary and not too sparse, so that our approach can better utilize and propagate relationships of dataset images to achieve large improvement. Due to different scaling of similarity scores from different features, it is difficult to directly determine weights for the affinity matrices. We instead adopt a probabilistic approach based on the mixture Markov model inspired by [42]. The model is essentially a random walk on multiple graphs. Note that Harel and Koren [43] also adopts random walk to cluster spatial data, but on a single graph rather than multiple graphs. Suppose a walker is at vertex $v_i \in V$ in graph G_m. In the next step, it has 1) a certain probability $p_m(v_i)$ of staying in the same graph G_m and then walks to another vertex v_j in this graph with transition probability $p_m(v_j|v_i)$, or 2) probability $p_{m'}(v_i)$ of switching to graph

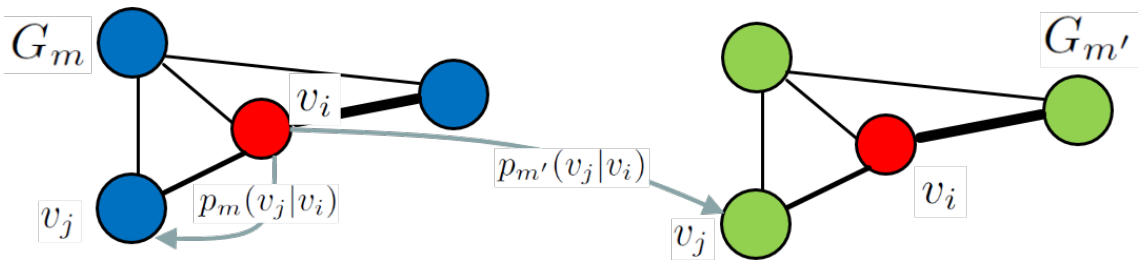


Figure 2.2: Illustration of mixture Markov model on two graphs.

$G_{m'}$ and then walks from v_i to v_j in graph $G_{m'}$ with transition probability $p_{m'}(v_j|v_i)$. Intuitively, sitting at a vertex, the walker first decides which graph to land in, jumps to that graph (or stays in the same graph), and then decides which neighboring vertex to go to according to the graph's affinity matrix. Mathematically, this procedure of walking from v_i to v_j across all graphs can be represented as

$$p(v_j|v_i) = \sum\nolimits_m p_m(v_j|v_i)p_m(v_i), \tag{2.1}$$

where $p(v_j|v_i)$ is the transition probability of walking from v_i to v_j in the fused graph. $p_m(v_i)$ is the probability of switching to (or staying in) graph G_m when the walker is at vertex v_i. It is the probability of switching between graphs. An intuitive illustration is presented in Figure 2.2.

Our next task is to compute the transition probability $p(v_j|v_i)$. Intuitively, $p(v_j|v_i)$ should be related to the edges between v_i and its neighbors. We resort to "degree of a vertex" and "volume of a graph" to explain our approach of computing $p(v_j|v_i)$. The degree of v_i in G_m is the sum of edge strength of all vertices connected to v_i, $d_m(v_i) = \sum_j e_m(v_i, v_j)$. The volume of graph G_m is the sum of all edge strength in the graph, $vol_m V = \sum_{v_i,v_j \in V} e_m(v_i, v_j) = \sum_{v_i \in V} d_m(v_i)$. The transition

probability is then written as

$$p_m(v_j|v_i) = e_m(v_i, v_j)/d_m(v_i). \tag{2.2}$$

After a number of steps, the random walk model will reach a stationary state where the stationary probability at vertex v_i is defined as

$$\pi_m(v_i) = d_m(v_i)/vol_m V. \tag{2.3}$$

Suppose the stationary probability of the fused graph is a linear combination of stationary probabilities of all graphs, $\pi(v_i) = \sum_m w_m(v_i)\pi_m(v_i)$, where $w_m(v_i)$ is the weight for vertex $v_i \in V$ in graph G_m, $w_m(v_i) \leq 1$ and $\sum_m w_m(v_i) = 1$. For a node in a graph, higher stationary probability implies higher probability of switching to (or staying in) this graph, so that $p_m(v_i) \propto \pi_m(v_i)$. Without other prior knowledge, we can estimate the probability $p_m(v_i)$ by linearly weighting the ratio of the stationary probability of an individual graph to that of the fused graph as

$$p_m(v_i) = w_m(v_i)\frac{\pi_m(v_i)}{\pi(v_i)}. \tag{2.4}$$

Plugging (2.2), (2.3) and (2.4) into (2.1), we obtain

$$p(v_j|v_i) = \frac{1}{\pi(v_i)}\sum\nolimits_m w_m(v_i)\frac{e_m(v_i, v_j)}{vol_m V}. \tag{2.5}$$

We introduce the edge strength between vertices v_i and v_j in the fused graph as

$$e(v_i, v_j) = \sum\nolimits_m w_m(v_i) \frac{e_m(v_i, v_j)}{vol_m V} \tag{2.6}$$

and obtain $p(v_j|v_i) = e(v_i, v_j)/\pi(v_i)$. The volume of the fused graph is 1. The affinity matrix of the fused graph is not symmetric due to the use of transition probability (the transition probabilities from v_i to v_j and v_j to v_i may not be the same). So $e(v_i, v_j)$ can be regarded as the weight of a directed edge. The mixture Markov model on the undirected graphs reduces to a convex combination of normalized affinity matrices. Therefore, we normalize all affinity matrices $\mathbf{S}_m$ to $\mathbf{T}_m$ by $\mathbf{T}_m = \mathbf{S}_m/vol_m V$, and discuss how to determine the weight $w_m(v_i)$ for each v_i in graph G_m in the next section.

2.3.4 Feature Weight Calculation

To obtain the weight $w_m(v_i)$, we describe a probabilistic model to determine the query-specific weights which measure the importance of a feature for a particular query. Our model is based only on the statistics of data and does not require any learning.

For a query image $\mathcal{I}_i$, we let $\mathcal{P}$ be the set of images similar to $\mathcal{I}_i$, and let $\mathcal{Q}$ be the set of images which are dissimilar from $\mathcal{I}_i$. Given a similarity score $s_{i,j}^m$ of feature $\mathcal{M}_m$ (graph G_m), the likelihood of a retrieved image $\mathcal{I}_j$ belonging to $\mathcal{P}$ or $\mathcal{Q}$ is denoted as $p(\mathcal{I}_j \in \mathcal{P}|s_{i,j})$ and $p(\mathcal{I}_j \in \mathcal{Q}|s_{i,j})$. By Bayes' theorem, we have $p(\mathcal{I}_j \in \mathcal{P}|s_{i,j}^m) = p(s_{i,j}^m|\mathcal{I}_j \in \mathcal{P})p(\mathcal{I}_j \in \mathcal{P})/p(s_{i,j}^m)$ and $p(\mathcal{I}_j \in \mathcal{Q}|s_{i,j}^m) = p(s_{i,j}^m|\mathcal{I}_j \in$

$\mathcal{Q})p(\mathcal{I}_j \in \mathcal{Q})/p(s^m_{i,j})$. We define $\rho_m(i,j)$ as the ratio of $p(\mathcal{I}_j \in \mathcal{P}|s^m_{i,j})$ to $p(\mathcal{I}_j \in \mathcal{Q}|s^m_{i,j})$

$$\rho_m(i,j) = \frac{p(\mathcal{I}_j \in \mathcal{P}|s^m_{i,j})}{p(\mathcal{I}_j \in \mathcal{Q}|s^m_{i,j})} = \frac{p(s^m_{i,j}|\mathcal{I}_j \in \mathcal{P})p(\mathcal{I}_j \in \mathcal{P})}{p(s^m_{i,j}|\mathcal{I}_j \in \mathcal{Q})p(\mathcal{I}_j \in \mathcal{Q})} \tag{2.7}$$

where $p(\mathcal{I}_j \in \mathcal{P})$ and $p(\mathcal{I}_j \in \mathcal{Q})$ represent the marginal probability of image $\mathcal{I}_j$ being a similar image or a dissimilar image given a query image. The marginal probabilities can be obtained by prior knowledge or an estimation of the portion of similar images that should be returned given a specific query. For examples, if we know there are 10% similar images given a query, we set $p(\mathcal{I}_j \in \mathcal{P}) = 0.1$ and $p(\mathcal{I}_j \in \mathcal{Q}) = 0.9$.

To obtain $p(s^m_{i,j}|\mathcal{I}_j \in \mathcal{P})$ and $p(s^m_{i,j}|\mathcal{I}_j \in \mathcal{Q})$, we make the assumption that the similarity scores between two similar images and those between two dissimilar images come from different distributions. To proceed, we manually annotate a set of pairs of similar images from the dataset offline to obtain the similarity scores of similar images. Additionally, we compute similarity scores between dataset images and images from an unrelated dataset (selected from the Caltech-101 dataset [44]) to obtain the similarity scores between dissimilar images. We approximate the distributions of the two sets of similarity scores as Gaussian distributions, $\mathcal{N}_\mathcal{P} \sim (\mu_\mathcal{P}, \sigma^2_\mathcal{P})$ and $\mathcal{N}_\mathcal{Q} \sim (\mu_\mathcal{Q}, \sigma^2_\mathcal{Q})$. Note that we use a Gaussian assumption for simplicity and efficiency, and will show that it works well in our experiments. Other data fitting algorithms can be applied to better capture the underlying distributions at the cost

of efficiency. In this way, (2.7) can be rewritten as

$$\rho_m(i,j) = \gamma \frac{p(s_{i,j}^m|\mathcal{N}_\mathcal{P})}{p(s_{i,j}^m|\mathcal{N}_\mathcal{Q})} = \gamma \frac{\sigma_\mathcal{Q}\mathcal{K}_\mathcal{P}(s_{i,j}^m)}{\sigma_\mathcal{P}\mathcal{K}_\mathcal{Q}(s_{i,j}^m)}, \tag{2.8}$$

where $\gamma = \frac{p(\mathcal{I}_j \in \mathcal{P})}{p(\mathcal{I}_j \in \mathcal{Q})}$, $\mathcal{K}_\mathcal{P}(s_{i,j}^m) = exp(-(s_{i,j}^m - \mu_\mathcal{P})^2/\sigma_\mathcal{P}^2)$ and $\mathcal{K}_\mathcal{Q}(s_{i,j}^m) = exp(-(s_{i,j}^m - \mu_\mathcal{Q})^2/\sigma_\mathcal{Q}^2)$.

In practice, we do not compute $\rho_m(i,j)$ for every retrieved image $\mathcal{I}_j$. Instead, for a query image $\mathcal{I}_i$, we compute the mean of the K largest similarity scores as $\bar{s}_i^m$, which indicates how reliable this ranked list is regarding the query image $\mathcal{I}_i$. By substituting $s_{i,j}^m$ with $\bar{s}_i^m$ in (2.8), we have a query-specific confidence score $\rho_m(i)$ by (2.8), which is denoted as $\rho_m(v_i)$ with the graph representation. The query-specific weight of a query v_i in graph G_m is computed by $w_m(v_i) = \rho_m(v_i)/\sum \rho_m(v_i)$. In our work, the query-specific weight is only assigned to the query node in a graph. However, it is also applicable to non-query nodes, although there is no need to adjust fusion weights for non-query nodes as they are excluded during evaluation. For a non-query image v_j in graph G_m, we simply use equal weight $w_m(v_j) = 1/r$ for r features. Therefore, we obtain a weight vector

$$\mathbf{w}_m = (w_m(v_1), w_m(v_2), ..., w_m(v_n))^\top \tag{2.9}$$

computed from all vertices for each graph G_m. The normalized affinity matrix of

the fused graph $\mathbf{T}$ is subsequently calculated as

$$\mathbf{T} = \sum\nolimits_m \mathbf{diag}(\mathbf{w}_m) \cdot \mathbf{T}_m \tag{2.10}$$

where the i-th diagonal element in $\mathbf{diag}(\mathbf{w}_m) \in \mathbb{R}^{n \times n}$ corresponds to $w_m(v_i)$. This process is equivalent to assigning different weights for a row from different features when combining affinity matrices. Our approach does not assign a single weight for each feature, thereby capturing more query-dependent information from the similarity scores.

2.3.5 Diffusion Process

From the new affinity matrix $\mathbf{T}$ obtained in (2.10), we can directly infer a new ranking. Nevertheless, the results can be improved by applying a diffusion process to $\mathbf{T}$ to reduce noise. The basic idea is to propagate the similarity score of a vertex to its neighboring vertices until a stationary state is reached. Here we employ an iterative diffusion process for efficiency. Given $\mathbf{T}$, the transition matrix is defined as $\mathbf{P} = \mathbf{D}^{-1}\mathbf{T}$, where $\mathbf{D}$ is a diagonal matrix whose i-th diagonal element $d(i,i) = d(v_i)$, where $d(v_i)$ is the degree of vertex v_i in the fused graph. We build a matrix $\mathbf{W}^t = (\mathbf{f}_1^t \quad \mathbf{f}_2^t \quad \cdots \quad \mathbf{f}_n^t)^\top$, where $\mathbf{f}_i^t$ is a column vector indicating the probability of being at a vertex starting from vertex v_i after t steps. We employ the LCDP algorithm [41], which iteratively updates $\mathbf{W}^t$ by $\mathbf{W}^{t+1} = \mathbf{P}_K \mathbf{W}^t \mathbf{P}_K^\top$, where $\mathbf{P}_K$ is the transition matrix for the K-NN graph G_K built by only keeping similarity scores of each node and its K nearest neighbors. The edge strength $e(v_i, v_j) = 0$

Algorithm 1 Multi-feature Re-ranking with Diffusion

Input: r affinity matrices $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_r\}$ representing r graphs $\mathcal{G}$, the query image $\mathcal{I}_i$

Output: Re-ranked results for $\mathcal{I}_i$

1: **for** $m = 1$ **to** r **do**

2: Normalize $\mathbf{S}_m$ to $\mathbf{T}_m$ (Section 2.3.3);

3: Compute the mean of the K largest similarity scores from $\mathbf{T}_m$ for $\mathcal{I}_i$ as $\bar{s}_i^m$;

4: Compute query-specific confidence $\rho_m(v_i)$ by (2.8);

5: Compute the weight vector $\mathbf{w}_m$ in (2.9), where $w_m(v_i) = \rho_m(v_i)/\sum \rho_m(v_i)$ for the query node and $w_m(v_i) = 1/r$ for non-query nodes.

6: **end for**

7: Obtain the affinity matrix $\mathbf{T}$ of the fused graph by (2.10);

8: Apply diffusion process to $\mathbf{T}$;

9: Infer new ranks from $\mathbf{T}$ for the query $\mathcal{I}_i$ by sorting similarity scores of the row associated with query node.

if vertex v_j does not belong to the K-NNs of v_i, and $\mathbf{W}^0 = \mathbf{P}_K$. Details can be found in [41]. The diffusion terminates after a pre-defined number of iterations or if $\mathbf{W}$ does not change. The diffused matrix is used to re-rank retrieved images to obtain the final results by sorting diffused similarity values of the row associated with the query node. The entire procedure of our fusion approach is presented in Algorithm 1.

2.4 Experiments

In this section, we evaluate our algorithm on several image retrieval datasets and compare it with a few state-of-the-art approaches.

2.4.1 Datasets

We test our approach on four widely used datasets, which are the Holidays [6], UKbench [9], Oxford5k [1] and Paris6k [2] datasets. The Holidays dataset is composed of 1491 images labeled as 500 categories, where each category consists no more than 10 images of an object, such as buildings, famous landmarks and natural scenes. The first image in each category is used as query to search for database images containing the same object. For each query, the remaining 1490 images are considered as database images. Note that most images from the same category in the Holidays dataset are under slight viewpoint and illumination change, which is usually referred to as near-duplicate scenario. Therefore, the retrieval task is less challenging compared to other datasets.

The UKbench dataset contains 10200 images from 2550 categories (objects or natural scenes) with 4 images for each object or scene, taken under different viewpoints and lighting conditions. Images are ordered so that the first image from each category is used as query to retrieve the remaining 3 images of the same category. Compared with the Holidays dataset, images in the UKbench dataset exhibit more various pose and illumination changes.

The Oxford5k dataset consists of 5062 photos of famous Oxford landmarks.

Groundtruth is provided for 11 different landmarks, each of which has 5 queries, resulting in 55 queries, while the remaining images serve as database images. Due to significant viewpoint change amongst images of the same landmark, it is very challenging to retrieve and highly rank all similar images given a query. In addition, different landmarks may look similar in some cases, making accurate retrieval more difficult.

Similar to Oxford5k dataset, the Paris6k dataset contains 6412 photos of famous buildings in Paris, of which 55 photos serve as queries. The queries also consist of 11 different landmarks, each of which contains 5 images.

It should be noted that the query region of Oxford5k and Paris6k datasets is only part of an image, provided by groundtruth, which is different from Holidays and UKbench datasets, where the entire image is used as a query. The retrieval task is more challenging for Oxford5k and Paris6k datasets since the query regions and correctly matched regions in database images may take only a small portion of the entire image. Therefore, the large amount of background may introduce noise that makes successfully finding the correct matches difficult. Moreover, the viewpoint significantly changes across images, while images in Holidays and UKbench datasets are mostly near-duplicate.

2.4.2 Experimental Setup

We use 2 local features and 2 global features that are widely used in existing image retrieval systems. For local features, we use Hessian affine feature point

extractor and the 128-dimension SIFT descriptor [45] to compute BOW features. We use the visual words provided by [45] except on Holidays dataset where we train a 1M vocabulary by approximate k-means (AKM) [1]. Single assignment and tf-idf weighting are applied to construct BOW vectors. We adopt the 8192-dimension VLAD descriptor with signed square root (SSR), computed with 64 clusters provided by [15], For global features, we use GIST descriptor [46] and HSV color histograms. The GIST descriptor is 1192-dimension while the color histogram is 4000-dimension with 40 bins for H and 10 bins for S and V components.

We compute cosine similarity between two BOW vectors. For other features, we compute the Euclidean distance x_d between two feature vectors and convert it to a similarity score by $exp(-x_d/\sigma)$. Our algorithm is not sensitive to σ, as we will show in the experiments. For simplicity, we set $\sigma_{\mathcal{P}} = \sigma_{\mathcal{Q}} = 1$ and fix them throughout all experiments. The parameter K, denoting the number of neighboring vertices in the K-NN graph and the number of top largest similarity scores of a query, is set to 6 for Holidays and UKbench, and 40 for Oxford5k and Paris6k. The length of the short list of retrieved images L is set to 700 for Holidays and UKbench, and 5000 for Oxford5k and Paris6k. Similarity scores between dataset images are computed offline, while the scores between queries and dataset images are computed online during retrieval. Graphs $\mathcal{G}$ are constructed during reranking using computed similarity scores.

For evaluation metrics, we use N-S score [9] on UKbench dataset, which measures the recall of the top 4 retrieved images, and mean average precision (mAP) on other 3 datasets.

2.4.3 Comparison with Existing Approaches

First, we compare our method with a few existing approaches. The quantitative comparison is shown in Table 2.1. The baselines using individual features in our work are initial retrieval results from pairwise similarities without any other techniques, *i.e.*, spatial verification (SV), query expansion (QE), multiple assignment (MA) or weak geometric consistency (WGC), *etc.* However, most other approaches using a single feature rely on various additional improvements. In particular, we compare with [28] and [30] which also exploit multiple features to improve retrieval performance. We will show that our fusion algorithm greatly improves baselines' performance and outperforms state-of-the-art approaches even we only uses similarity scores. Note that we are not designing superior baselines, which is outside the scope of this work.

As shown in Table 2.1, the BOW representation achieves the best retrieval performance among all baselines across different datasets, while GIST and color features are not discriminative enough. Nevertheless, our multi-feature fusion algorithm significantly improves the final retrieval performance on all datasets and outperforms state-of-the-art algorithms. On Holidays and UKbench datasets, we obtain 88.3% mAP and 3.86 N-S score respectively, which are the best reported results to our knowledge. Compared to the best baseline (BOW), our fusion improves the results by 14.4% on Holidays and 10.3% on UKbench with a simple probabilistic model. In contrast, the relative improvements by [28] that is also based on graph fusion are 9.2% (77.5% to 84.6%) on Holidays and 6.5% (3.54 to 3.77) on UKbench,

Table 2.1: Comparisons with state-of-the-art approaches. We use N-S score on UKbench, and mAP (in %) on other datasets. "-" means the results are not reported. B, SV, MA, QE and WGC stand for baseline (single feature), spatial verification [1], multiple assignment [2], query expansion [3–5] and weakly geometric consistency [6].

	Methods	Holidays	UKbench	Oxford5k	Paris6k
Baseline	BOW [45]	77.2	3.50	67.4	69.3
	VLAD [15]	55.9	3.22	32.6	38.0
	GIST [47]	35.0	1.96	24.2	19.2
	Color	55.8	3.09	8.5	8.4
B+SV/MA/QE/WGC	Philbin *et al.* [1]	-	3.45	66.4	-
	Jégou *et al.* [6]	75.1	-	54.7	-
	Jégou *et al.* [24]	84.8	3.64	68.5	-
	Qin *et al.* [45]	-	3.67	81.4	80.3
	Chum *et al.* [3]	-	-	82.7	80.5
	Mikulik *et al.* [48]	75.8	-	**84.9**	82.4
	Qin *et al.* [49]	82.1	-	78.0	73.6
	Tolias *et al.* [50]	88.0	-	83.8	80.5
Fusion	Ours	**88.3**	**3.86**	76.2	**83.3**
	Zhang *et al.* [28]	84.6	3.77	-	-
	Zhang *et al.* [30]	80.9	3.60	68.7	-

while they are 9.6% (73.8% to 80.9%) and 5.4% (3.42 to 3.6) by [30]. Compared to other single feature based methods with sophisticated processing steps, our fusion depends only on similarity scores to calculate query-specific weights and perform

diffusion process, and exploits more reliable information about the relationships among images, thus producing better retrieval results.

On Oxford5k and Paris6k datasets, the color feature only achieves 8.5% and 8.4% mAP due to large viewpoint changes, cluttered background and a constrained region of interest (ROI) for query. Additionally, the performance of GIST and VLAD features also drops. Different from [28], we do not specifically remove an inferior feature, but include all features in the fusion without any bias, even though the color feature performs much worse than others. It is clear that our fusion still greatly improves final retrieval performance. Our experiments clearly shows that our fusion is very robust and is not deteriorated by a single inferior feature (color). It improves the best baseline (BOW) by 13.1% and achieves 76.2% mAP on Oxford5k, which outperforms [30] and is comparable to other state-of-the-art approaches. On Paris6k, our fusion brings the mAP from 69.3% by the best baseline (BOW) up to 83.3% without spatial verification, query expansion and other techniques, which is a 20.1% relative improvement. The performance gain is larger than that on the near-duplicate datasets where individual features have already achieved good performance due to less variance, making the potential of fusion limited. In contrast, on Oxford5k and Paris6k, a single feature is often not powerful enough to distinguish different images and multiple features better complement each other. An example of reranking result is shown in Figure 2.3.

Figure 2.3: An example of retrieved images by four features and our fusion method on Holidays dataset [6]. The left-most image is the query. Retrieved images are ranked higher if they have high similarity scores with the query. Images with red bounding boxes are correct matches.

2.5 Discussion and Analysis

In this section, we conduct further experiments to diagnose our approach and analyze the effect of its components, so that we can have a better understanding of its performance.

2.5.1 Contributions of Individual Components

We first evaluate the importance of individual components of the proposed method. We conduct additional experiments by adding or removing a component and measuring how accuracy changes. The configurations are detailed as follows. With the original affinity matrices from multiple features, the accuracy can be measured by selecting the maximal mAP among all baselines, denoted as B. The ap-

proaches by fusion with equal weights and query-specific weights are denoted as EW and QW, respectively, where results are directly inferred from the combined affinity matrix without diffusion. Both the EW and QW approaches use all dataset images. Two variants using a short list are denoted as SL+QW and SL+EW. Our entire framework is denoted as SL+QW+DP, while the variant using EW and SL for diffusion is denoted as SL+EW+DP. The comparisons on the test datasets are shown in Table 2.2.

We can see that both QW and DP contribute to the improvements while using a proper SL also increases accuracy. Specifically, in most cases, results by QW are better than those by EW, showing the effectiveness of our probabilistic model derived from statistics of similarity scores. Additionally, if there are a large number of relevant images to be retrieved for a query (Oxford5k and Paris6K), we need to include more images in the short list to obtain good results; otherwise the performance drops below the best baseline because many similar images are excluded from the fused graph. In contrast, a small short list is sufficient when there are only a few similar images to be retrieved. Therefore, we can control the length of short list to achieve a trade-off between computational complexity and accuracy.

2.5.2 Parameter Evaluation

The proposed method has several parameters to set: the length of the short list L, the number of nearest neighbors K in K-NN graph and σ for converting the Euclidean distance to similarity score for VLAD, GIST and color features. To

Table 2.2: Retrieval performance by different variants of the proposed method. N-S score on UKbench, and mAP (in %) on other datasets.

Methods	SL length L	Holidays	UKbench	Oxford5k	Paris6k
B	-	77.2	3.50	67.4	69.3
EW	-	81.1	3.72	69.2	68.1
QW	-	84.0	3.76	70.3	71.2
SL+EW	700	82.1	3.76	63.7	65.7
	1500	-	3.76	64.3	66.0
	5000	-	3.75	69.1	67.6
SL+QW	700	83.6	3.77	65.6	67.5
	1500	-	3.77	65.3	68.9
	5000	-	3.77	70.3	69.6
SL+EW+DP	700	86.4	3.84	73.2	80.1
	1500	-	3.84	73.8	80.8
	5000	-	3.84	74.0	81.4
SL+QW+DP	700	**88.3**	**3.86**	75.2	82.0
	1500	-	3.86	75.7	82.6
	5000	-	3.85	**76.2**	**83.3**

evaluate the sensitivity of our method to these parameters, we conduct experiments by changing one parameter at a time. The retrieval results regarding different L are shown in Table 2.2. Performance by changing other parameters are shown in Figure 2.4.

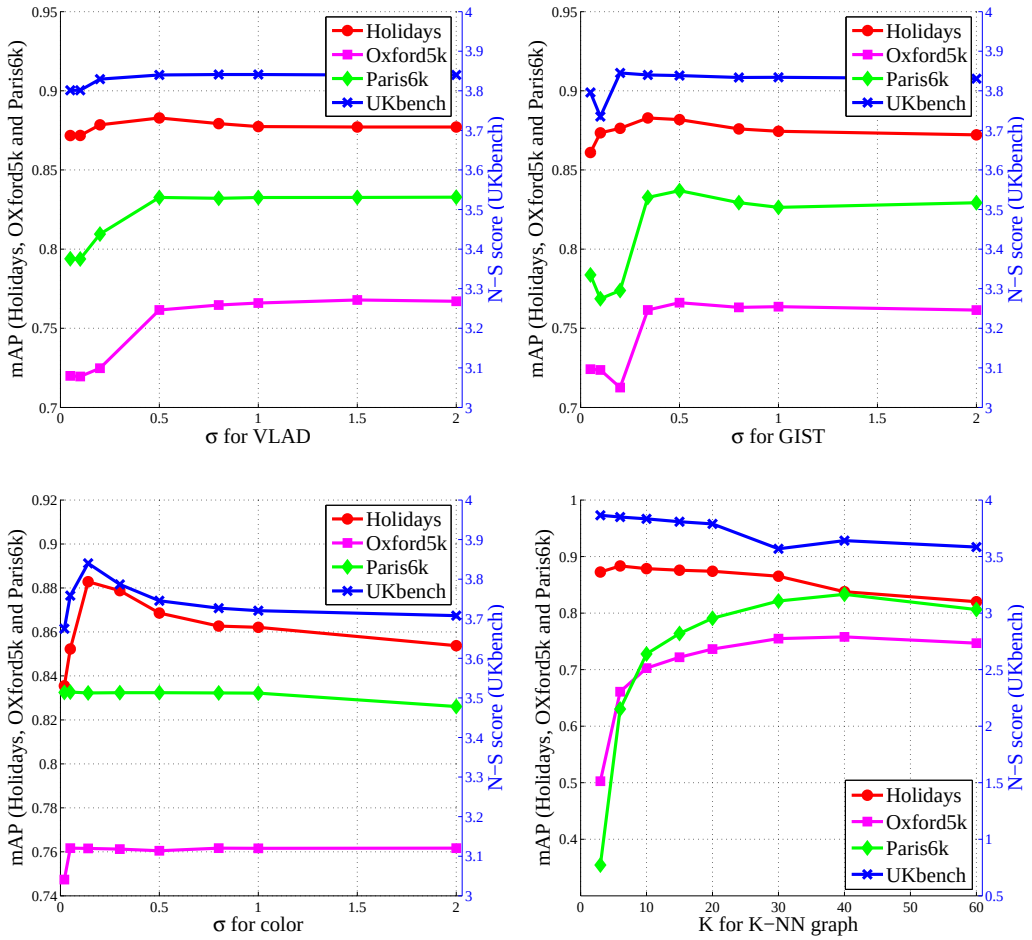


Figure 2.4: Performance under different σ for VLAD, GIST and color, and K for K-NN graph used in the diffusion process.

Our method is robust and not sensitive to these parameters as long as they are in a reasonable range. In particular, performance does not change much even when σ is 4 times of its optimal value, meaning that we can safely fix a larger σ for all datasets without sacrificing accuracy too much. In all experiments, σ is empirically set to 0.5, 0.34 and 0.14 for VLAD, GIST and color features. Additionally, on Oxford5k and Paris6k datasets which consist of a large number of similar images for each query, we need a large K to include them in the graph and highly rank them

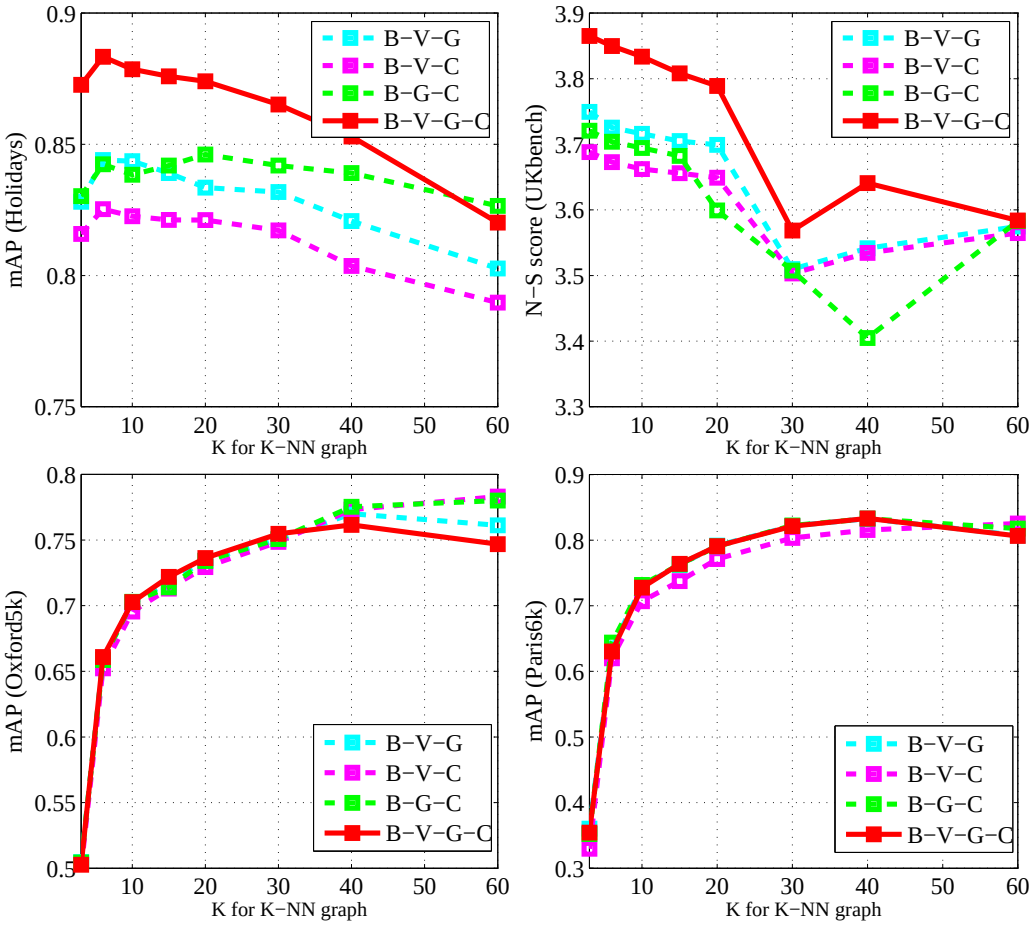


Figure 2.5: Performance of different feature combinations with respect to varying K. B, V, G and C stand for BOW, VLAD, GIST and color features.

after re-ranking. In contrast, on Holidays and UKbench datasets which only contain a small number of similar images, a small K is sufficient to include most of them in the graph; otherwise similarity scores of those similar images will be contaminated by irrelevant images if K is too large.

2.5.3 Combinations of Features

We conduct experiments using different feature combinations to further verify the effectiveness of our fusion algorithm. In Figure 2.5, we show the perfor-

mance using 4 combinations of features which fuse 3 or 4 types of features. Since VLAD+GIST+color performs much worse than other combinations, we do not display its results in the figure for better visualization [1]. In most cases, fusing features from all 4 features achieves the best results, especially on Holidays and UKbench datasets, which verifies that our fusion algorithm is very robust and is not easily affected by an inferior feature (color in this case). Moreover, our fusion successfully exploits complementary information from multiple features, thereby greatly improving the performance compared to combinations of 3 features. Only when K becomes very large, the performance by fusing all 4 features is slightly worse than that by other combinations due to large amount of noise from multiple features. Note that our fusion does not set any restrictions on the number or type of features to be fused.

2.6 Summary

In this Chapter, we have introduced an image reranking algorithm by multi-feature fusion with diffusion for image retrieval. We exploit the pairwise similarity scores between images to infer their relationships. Initial ranks from one feature are represented as an undirected graph where edge strength is similarity score. Graphs are combined by a mixture Markov model where the query-specific weight is calculated by a probabilistic model utilizing the statistics of similarity scores. Diffusion is then applied to the fused graph to reduce noise. Our approach significantly and

[1]The best results by VLAD+GIST+color are 52.4%, 2.91, 30.5% and 40.3% on Holidays, UKbench, Oxford5k and Paris6k datasets.

consistently improves the performance of baselines and is very robust to variations in its parameters.

Chapter 3: Image Retrieval by Submodular Reranking

3.1 Overview

In the previous chapter, we have introduced a supervised reranking algorithms to improve initial retrieval results from multiple features, where a set of irrelevant images are manually annotated for computing the combination weights of features. However, the supervised approach is not feasible enough when there is no annotation available. Moreover, it is time-consuming to collect a large set of images and impractical to ensure they are irrelevant to the database images if the retrieval database is already very large.

To address the aforementioned drawbacks, we attempt to reduce the effort of human labeling and propose an unsupervised retrieval algorithm in this chapter. Given initial ranked lists from multiple features, we only utilize the pairwise image similarities of the query and initially retrieved database images without any supervised information. Similar to the proposed approach in Chapter 2, this approach is also based on graph representations of initial retrieval results. In short, we formulate the reranking problem as selecting and rearranging a subset of retrieved images from initial ranked lists obtained from multiple features. We further cast the subset selection problem as optimizing an objective function that is constructed as a

submodular and non-decreasing function. Our submodular objective function utilizes similarities of pairs of images to exploit relationships between retrieved images within each feature. It also considers the relative ranking between retrieved images across multiple ranked lists. Due to the *diminishing returns property* of submodular functions, the optimization can be efficiently solved by simple greedy algorithm with performance guarantee.

3.2 Related Work

Since we have already discussed a few previous works on multi-feature fusion for image retrieval in Chapter 2, we will focus on submodular optimization and classic reranking algorithms in this section.

3.2.1 Submodular Optimization

Submodularity, as a discrete analog of convexity, is widely studied in combinatorial optimization [51] due to its diminishing returns property: adding an element to a smaller set contributes more than adding it to a larger set. It is initially applied to machine learning tasks to solve complicated optimization problems efficiently. Later on, various submodular functions have been proposed and successfully applied to many vision applications, such as image segmentation [52, 53], superpixel segmentation and clustering [54, 55], dictionary selection/learning [56, 57], saliency detection [58], object recognition [59] and video hashing [60]. A few works applied submodular functions to diversified ranking [61–63], where elements in the reranked

list are similar to the query but also diversified. For diversified ranking, submodular functions are designed to seek a trade-off between similarity and diversity. It should be noted that [61–63] are not similar to our submodular reranking, since we encourage elements in the reranked list to be similar to the query and homogeneous rather than diversified.

3.2.2 Image Reranking

For image reranking, [64] proposed a click boosting method using the user click data to help rerank initially retrieved images by textual and visual features, which may not be applicable when click data is missing. Voravuthikunchai *et al.* [65] proposed to mining frequent closed patterns as image representations, and designed a scoring function to rerank images using mined patterns. Yu *et al.* [66] adopted a hypergraph-based sparse coding algorithm to predict clicks using multiple visual features. An initial ranked list is reranked based on predicted clicks of retrieved images. Multi-feature fusion is also widely used in image retrieval. Wang *et al.* [27] designed a graph-based learning algorithm for inferring weights of features, which requires a large number of queries beforehand to estimate relevance scores of initially retrieved images. Similarly, Chavez *et al.* [67] utilized a Markov random field and manual relevance feedback to combine retrieval results by visual and textual features.

3.3 Proposed Approach

3.3.1 Preliminaries

Before introducing our approach, we would like to explain a few definitions regarding submodularity and monotonicity to help understanding the formulation of our proposed approach.

Submodularity. Let $\mathcal{V}$ be a finite set. A set function $f : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is submodular if it satisfies $f(\mathcal{S} \cup a) - f(\mathcal{S}) \geq f(\mathcal{T} \cup a) - f(\mathcal{T})$ for all $\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{V}, a \in \mathcal{V} \backslash \mathcal{T}$. This is called the *diminishing returns property*: adding a to a small set has a bigger impact than adding it to a larger set. The gain of the function value $f(\mathcal{S} \cup a) - f(\mathcal{S})$ is called the *marginal gain* of f when adding a to $\mathcal{S}$.

Monotonicity. A set function $f : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is monotone (or non-decreasing) if for every $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{V}$, $f(\mathcal{S}) \leq f(\mathcal{T})$ and $f(\emptyset) = 0$.

3.3.2 Information Gain with Graphical Models

Given M features, we obtain M initial ranked lists of retrieved images for each query image. For efficient reranking, we select only the top K retrieved images from each ranked list. Note that the top K images are generally not the same across different features. Given an initial ranked list consisting of K retrieved images from feature m, we represent it as an undirected graph $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ where nodes $v_m \in \mathcal{V}_m$ are images and $e_m(i, j) \in \mathcal{E}_m$ denotes the edge that connects $v_m(i)$ and $v_m(j)$ (see Figure 3.1). An affinity matrix $\mathbf{A}_m \in \mathbb{R}^{K \times K}$ is used to represent the graph

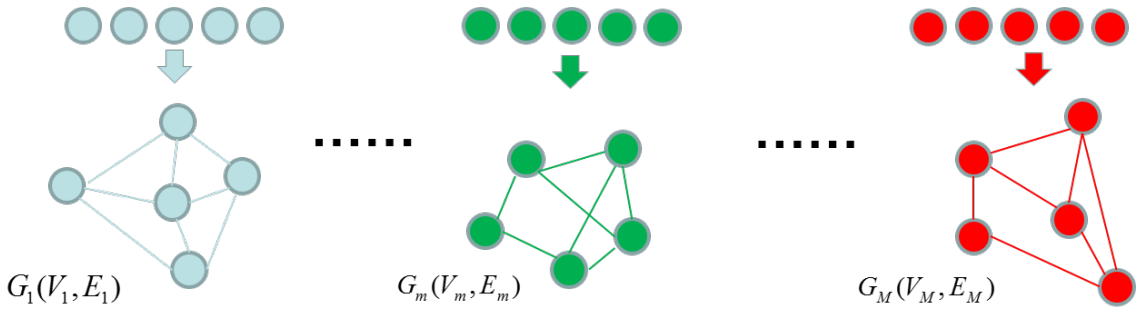


Figure 3.1: Graph representations of multiple ranked lists.

with the element $a_m(i,j)$ corresponding to the edge weight of $e_m(i,j)$, which is the pairwise similarity between images $v_m(i)$ and $v_m(j)$[1]. To facilitate the objective function construction (see Section 3.3.2), we do not include self-loops $e_m(i,i)$ of nodes $v_m(i)$ in the graph. Therefore, $a_m(i,i)$ is set to 0. For notational convenience, we denote $\mathcal{V}$ as the union of all nodes from the M undirected graphs, so that $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \cdots \cup \mathcal{V}_M$. We aim to select a subset of nodes $\mathcal{S}$ from $\mathcal{V}$ which are the most similar to the query image and arrange them in order to obtain the reranked result. Furthermore, $\mathcal{U}$ denotes the set of images which are not selected, so that $\mathcal{U} \cap \mathcal{S} = \emptyset$ and $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$.

Given M graphs, we seek a method to combine them so that complementary features may help discover images similar to the query in a joint manner. Although the same graph construction is used for all ranked lists, pairwise similarities from different features are usually of incomparable scales, making a direct graph combination infeasible. To address this problem, we resort to information gain theory with graphical models [68], which is based on a simple probabilistic model.

We start from the random walk model on a graph $\mathcal{G}_m$. The random walk model

[1]Please see experiment section about how to compute pairwise similarities.

can be interpreted as a Markov process: a walker stays at a node in the graph at time t and randomly walks to one of its neighboring nodes under some probability at time $t+1$. The probability of "walking" between nodes is called the transition probability and is defined as $\mathbf{P}_m = \mathbf{D}_m^{-1}\mathbf{A}_m$, where $\mathbf{D}_m \in \mathbb{R}^{K\times K}$ is a diagonal matrix with the diagonal element $d_m(i,i) = \sum_j a_m(i,j)$. The transition matrix $\mathbf{P}_m$ is a row-stochastic matrix indicating the transition probabilities of a random walk on the graph. $p_m(i,j)$ represents the conditional probability of walking from node $v_m(i)$ to node $v_m(j)$, which indicates the similarity between $v_m(i)$ and $v_m(j)$ based on the observation of $v_m(i)$. With the transition matrix $\mathbf{P}_m$, edge weights are converted to probabilities. Then we adopt information gain as a direct measure of the value of information of our graphical models. We start from a single graph $\mathcal{G}_m$, and define the information gain as

$$F_m(\mathcal{S}) = H(\mathcal{V}_m \backslash \mathcal{S}) - H(\mathcal{V}_m \backslash \mathcal{S}|\mathcal{S}) \tag{3.1}$$

where $\mathcal{S}$ is the subset we select from $\mathcal{V}$, and $\mathcal{V}_m \backslash \mathcal{S}$ is the set $\mathcal{V}_m$ with $\mathcal{S}$ removed. $H(\mathcal{V}_m \backslash \mathcal{S})$ is the entropy of unselected nodes in graph $\mathcal{G}_m$. $H(\mathcal{V}_m \backslash \mathcal{S}|\mathcal{S})$ is the conditional entropy of remaining nodes on graph $\mathcal{G}_m$ after we have observed $\mathcal{S}$. Specifically, $H(\mathcal{V}_m \backslash \mathcal{S}|\mathcal{S})$ and $H(\mathcal{V}_m \backslash \mathcal{S})$ are defined as

$$\begin{aligned} H(\mathcal{V}_m \backslash \mathcal{S}|\mathcal{S}) &= -\sum_{v\in\mathcal{V}_m\backslash\mathcal{S}, s\in\mathcal{S}} p_m(v,s)\log p_m(v|s) \\ H(\mathcal{V}_m \backslash \mathcal{S}) &= -\sum_{v\in\mathcal{V}_m\backslash\mathcal{S}} p_m(v)\log p_m(v) \end{aligned} \tag{3.2}$$

where $p_m(v,s) = p_m(v|s)p_m(s)$. $p_m(v|s)$ is the transition probability of walking to a node v in graph $\mathcal{G}_m$ when the walker is at node s. $p_m(s)$ and $p_m(v)$ are the marginal probabilities of nodes s and v being similar to the query from feature m. $p_m(v|s)$ can be directly obtained from $\mathbf{P}_m$. To calculate the marginal probability $p_m(v)$, we use the normalized similarities between the query and retrieved images. We denote the similarities between the top K retrieved images and the query image from feature m as $\mathbf{c}_m = (c_{m,1}, c_{m,2}, ..., c_{m,K})^\top$. ℓ_1 normalization is then applied to $\mathbf{c}_m$ to obtain $p_m(v) = c_{m,v}/|\mathbf{c}_m|_1$.

We have the following proposition stating that the information gain with our graphical model is submodular.

Proposition 1. *$F_m : 2^{\mathcal{V}_m} \to \mathbb{R}$ is a submodular and monotone function.*

The proof is presented in the Appendix. F_m is essentially the mutual information $I(\mathcal{V}_m \backslash \mathcal{S}; \mathcal{S})$ capturing the mutual dependence between subset $\mathcal{S}$ and unselected nodes $\mathcal{V}_m \backslash \mathcal{S}$, which measures how much $\mathcal{S}$ is representative of the graph with respect to the query. That F_m is non-decreasing is obvious, because the addition of any node to $\mathcal{S}$ always provides information or does not provide information at all, since "information never hurts". Submodularity comes from the observation that the information gain of adding a node to $\mathcal{S}$ becomes less in a later stage because it is more likely similar to elements in $\mathcal{S}$ as $\mathcal{S}$ grows.

To combine graphs, we need to determine the importance of each graph. Here we adopt the heuristic of simply summing up the information gains of the individual

graphs to obtain the total information gain:

$$R(\mathcal{S}) = -\sum_{m}\Big(\sum_{v\in\mathcal{V}\setminus\mathcal{S}} p_m(v)\log p_m(v) - \sum_{v\in\mathcal{V}\setminus\mathcal{S},s\in\mathcal{S}} p_m(v,s)\log p_m(v|s)\Big) \tag{3.3}$$

The information gain on a graph takes relationships between dataset images into account, so it propagates information about a dataset image to its neighbors, and better exploits dataset images that are similar to the query than simple pairwise comparisons. The combination seeks an agreement with respect to pairwise similarities derived from multiple features, so explores relationships of features to some extent. Note that since the top K images retrieved from different features may not be the same, $p_m(v)$ and $p_m(v|s)$ are set to 0 if an image is not included in graph $\mathcal{G}_m$, so it does not contribute to the objective function. An image discovered by most features contributes more to the information gain, therefore is selected to be in $\mathcal{S}$ with greater chance.

Since $F_m(\mathcal{S})$ is submodular and monotonically increasing, the linear combination of submodular functions, $R(\mathcal{S})$, is also submodular and non-decreasing. Since the information gain exploits the pairwise relationships between retrieved images, maximizing $R(\mathcal{S})$ is equivalent to selecting a group of images that are similar to the query and closely related to each other. Intuitive examples are shown in Figure 3.2. The number next to the edges is weight (similarity) between nodes. The marginal probability of all nodes is set to 1/4. Four cases of selection are presented, where the corresponding value of $F_m(\mathcal{S})$ is shown under each sub figure. By computing the information gain, we observe that it prefers images that are closely related to each

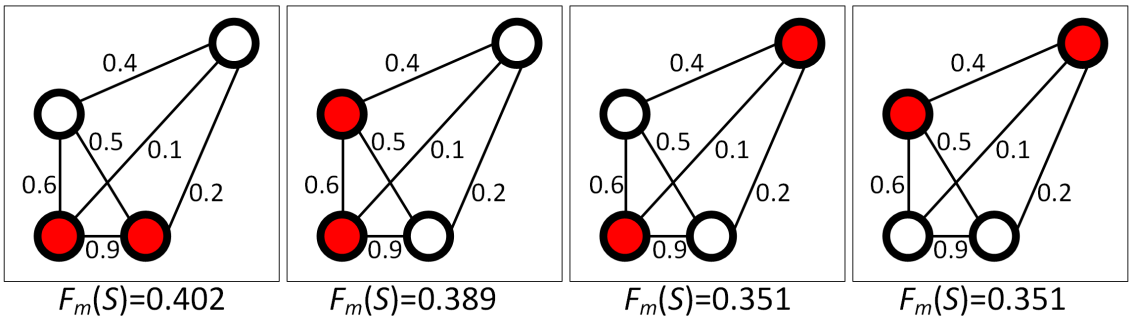


Figure 3.2: The importance of information gain for selecting nodes into subset $\mathcal{S}$. Red dots represent the selected subset $\mathcal{S}$ while white dots are remaining nodes $\mathcal{V}_m \backslash \mathcal{S}$.

other to be selected into $\mathcal{S}$, resulting in a compact cluster. Therefore, relationships of dataset images are exploited to facilitate reranking.

3.3.3 Relative Ranking Consistency

Simply summing up initial ranks obtained from different features for an image is not suitable, as a higher rank may be overly diluted by other lower ranks. Although complementary information from multiple features is used by integrating the $F_m(\mathcal{S})$, information gain does not completely utilize the inter-relationships between features. Additionally, it only considers pairwise similarities between images. However, the initial ranks of retrieved images from different features provide additional information that can further improve performance. For example, an image that is similar to the query and ranked lower by one feature may be ranked higher when it is perceived from a different perspective (*i.e.*, different feature). We propose a simple yet effective relative ranking consistency measure to model inter-relationships of multiple ranked lists.

Our measure is based on two criterion. First, relationships of relative ranks

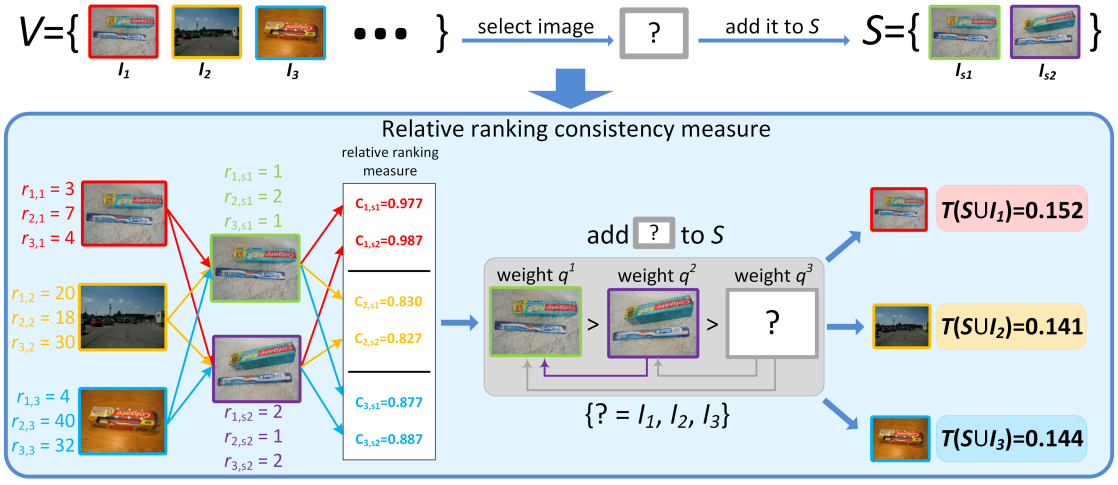


Figure 3.3: The effectiveness of the relative ranking consistency measure. See text for details.

between retrieved images should be maintained. Images with similar ranks in the initial ranked lists from different features should also be ranked closely after reranking. Second, images with consistent ranks across multiple features should have their ranks preserved after reranking. An image that is similar to the query but highly ranked by only a smaller number of features should also be captured. In contrast to the information gain term, this relative ranking consistency measure models inter-relationships of features at a higher level: using ranks themselves rather than pairwise similarities between images.

Again, as in Section 2.3.2, we only consider the top K images from each ranked list and denote $\mathcal{V}$ as the union of all retrieved images. Our goal is to select a subset of retrieved images $\mathcal{S} \subseteq \mathcal{V}$. We first define the *relative ranking* between a pair of images and then use it to measure the "inter-rank" consensus amongst multiple ranked lists.

Let $\mathbf{r}_m \in \mathbb{R}^K$ denote the positions of the top K images in the initial ranked list by feature m, $\mathbf{r}_m = (r_{m,1}, r_{m,2}, ..., r_{m,K})^\top$, where $r_{m,i}$ is the position of image I_i in the m-th ranked list. Smaller value means higher rank. The relative ranking between two images is defined as

$$rr_m(v_i, v_j) = |r_{m,v_i} - r_{m,v_j}|, \quad v_i, v_j \in \mathcal{V} \tag{3.4}$$

where v_i and v_j correspond to images I_i and I_j in the graph representations. If either v_i or v_j is not included in the top K images by feature m, $rr_m(v_i, v_j)$ is set to K. The relative ranking considers the difference between ranks of retrieved images. Similarly, for feature m', we also have the relative ranking, $rr_{m'}(v_i, v_j)$, of the same image pair in a different feature. On the one hand, the consensus between $rr_m(v_i, v_j)$ and $rr_{m'}(v_i, v_j)$ indicates that the rank relationship between v_i and v_j is reliable and should be maintained after reranking, which is related to the "consistency" between ranked lists. On the other hand, we also aim to discover images which are similar to the query but highly ranked by only a small number of features, thereby capturing the "distinctiveness" of specific features. To enforce both consistency and distinctiveness constraints, we define a relative ranking consistency measure across multiple ranked lists as

$$\mathcal{C}(v_i, v_j) = \frac{1}{Z} \sum_{m,m' \in M, m \neq m'} 1 - \frac{\min(rr_m, rr_{m'})}{K} \tag{3.5}$$

where $Z = \frac{M(M-1)}{2}$ is a normalization factor corresponding to the number of all

possible feature pairs. With this measure, if images I_i and I_j are ranked similarly across multiple features, they will also have similar ranks in the reranked list, *i.e.*, they both will be selected and highly ranked in $\mathcal{S}$ or both will be excluded from $\mathcal{S}$. This results from the constraint on relative ranking consistency. Now consider the situation in which an image I_i is ranked closely to a visually similar image I_j only in a small number of features. In this case, we still discover such similarity due to the use of the min function, and rank these images appropriately. If either v_i or v_j is not included in the top K images by features m and m', $1 - \frac{\min(rr_m, rr_{m'})}{K} = 0$, which indicates that these two images have disparate ranks and should contribute nothing to the objective function. Therefore, we take the inter-relationships amongst multiple ranked lists into account with respect to the relative ranking between two images. Several examples are shown in Figure 3.3. In Figure 3.3, the set $\mathcal{V}$ contains $K = 100$ images, from which we need to select an image into $\mathcal{S}$, which currently contains two images. Starting from initial ranks from the three features, we compute the relative ranking consistency measure between images in $\mathcal{V}$ and $\mathcal{S}$. For illustration purposes, we only show the values of the relative ranking consistency measure for 3 images (I_1, I_2 and I_3) in the set $\mathcal{V}$. I_1 in $\mathcal{V}$, which is initially ranked close to images in $\mathcal{S}$ across all features, has the largest relative ranking consistency $\mathcal{C}$. The relative ranking consistency of I_3, which is highly ranked by only a single feature, is larger than that of I_2 in $\mathcal{V}$, which is lower ranked by all features. Therefore, the relative ranking consistency term favors adding I_1 to $\mathcal{S}$ as it produces the largest function value for $T(\mathcal{S})$. Then it favors adding I_3 over I_2, which has the smallest function value. Our relative ranking consistency successfully captures inter-relationships amongst

multiple ranked lists and uses them to select images.

Finally, we define a set function based on the rank biased overlap (RBO) similarity [69], incorporating the aforementioned relative ranking consistency measure. RBO similarity was proposed in [69] but they did not observe or take advantage of its submodularity property. We extend the basic idea from [69] that highly ranked images should be more important than lower ranked images in our objective function. Suppose the images in $\mathcal{S}$ are ordered and that the position of image I_i in the new ranked list is r_{v_i}. The relative ranking consistency term is defined as

$$T(\mathcal{S}) = (1-q)\sum_{s=1}^{|\mathcal{S}|} q^s \cdot \frac{1}{s}\sum_{v_i,v_j\in\mathcal{S},r_{v_i}<r_{v_j}=s} \mathcal{C}(v_i,v_j) \tag{3.6}$$

where the term $\frac{1}{s}\sum_{v_i,v_j\in\mathcal{S},r_{v_i}<r_{v_j}=s}\mathcal{C}(v_i,v_j)$ allows us to select the image v_j with new rank s and compute the average relative ranking measure between v_j and all other s images with higher new ranks than v_j (see Figure 3.3). $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$. With the requirement that highly ranked images should have more weight in the objective function than lower ranked images, we introduce a weight parameter q for each image according to its new rank in $\mathcal{S}$. q controls the steepness of weight decay, so that a higher ranked image contributes more to the function value. Starting from the top ranked image with $s = 1$, the function assigns weight q^s to this image v_j and iteratively computes the average relative ranking between v_j and other higher ranked images v_i ($r_{v_i} < r_{v_j}$). Maximizing this function leads to a subset of images $\mathcal{S}$, where images are highly ranked and similarly ranked with each other in the initial ranked list. Since at least two images are needed to compute the relative ranking

consistency measure, a phantom item v_p is included into $\mathcal{S}$ to select the first image. In practice, we use the query itself as the phantom with rank $r_{v_p} = 0$. Then we have the following proposition with the proof in the Appendix.

Proposition 2. *$T : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is a submodular and monotone function if elements in $\mathcal{S}$ are ordered with respect to a phantom item $v_p \in \mathcal{S}$ and $r_{v_p} = 0$.*

3.3.4 Optimization

Combining the information gain and relative ranking consistency terms, we obtain the final objective function $Q(\mathcal{S}) = R(\mathcal{S}) + \lambda T(\mathcal{S})$ for the reranking problem. The solution is obtained by maximizing the objective function:

$$\begin{aligned} \max_{\mathcal{S}} \quad & R(\mathcal{S}) + \lambda T(\mathcal{S}) \\ s.t. \quad & \mathcal{S} \subseteq \mathcal{V}, |\mathcal{S}| \leq K_s \end{aligned} \tag{3.7}$$

where λ is a pre-defined weighting factor balancing the two terms. K_s is the largest number of selected images, which means we only select and rerank at most K_s images. (3.7) is submodular and non-decreasing since it is a linear combination of submodular and non-decreasing functions. Direct optimization of (3.7) is an NP-hard problem, but it can be approximately optimized by a greedy algorithm. Starting from an empty set $\mathcal{S} = \emptyset$, the greedy algorithm iteratively adds a new element to $\mathcal{S}$ which provides the largest marginal gain at each iteration, until K_s elements have been selected. Specifically, during each iteration, we search for an image $a^* \in \mathcal{V} \backslash \mathcal{S}$, which gives the largest combined marginal gain from the infor-

Algorithm 2 Submodular Reranking

Input: Graphs $\{\mathcal{G}_1, ..., \mathcal{G}_M\}$, initial ranked lists $\{\mathbf{r}_1, ..., \mathbf{r}_M\}$, K_s and λ.

Output: Reranked list $\mathbf{r}$ and final retrieved images $\mathcal{S}$.

Initialize $\mathcal{S} \leftarrow \emptyset$, $\rho^{cur} \leftarrow 0$, $\mathbf{r} \leftarrow \mathbf{0}$;

while $|\mathcal{S}| < K_s$ **do**

  $a^* = \underset{\mathcal{S} \cup \{a\} \in \mathcal{V}}{\arg\max}\, Q(\mathcal{S} \cup \{a\}) - Q(\mathcal{S})$;

  if $Q(\mathcal{S} \cup \{a^*\}) \leq Q(\mathcal{S})$ **then**

    break;

  end if

  $\rho^{cur} \leftarrow \rho^{cur} + 1$;

  $\mathcal{S} \leftarrow \mathcal{S} \cup \{a^*\}$; $r_{a^*} \leftarrow \rho^{cur}$;

end while

mation gain and relative ranking consistency terms, add it to $\mathcal{S}$ and set its rank to $r_{a^*} = \rho^{cur}$, where ρ^{cur} indicates the iteration step. The iteration terminates when $|\mathcal{S}| = K_s$. The reranked images are those from $\mathcal{S}$, and ranks are also obtained. We can tune K_s to control the efficiency and accuracy of the algorithm. The entire process is presented in Algorithm 2. The constraint on the number of reranked images leads to a uniform matriod $\mathcal{M} = (\mathcal{V}, \mathcal{I})$, where $\mathcal{I}$ is the collection of subsets $\mathcal{S} \subseteq \mathcal{V}$ satisfying the constraint that the number of reranked images is less than K_s. Maximizing a submodular function with a uniform matriod constraint yields a $(1 - 1/e)$ approximation to the optimal solution [51].

To further accelerate the optimization, we adopt lazy evaluation [57] to avoid

recomputing the function value for each node $a^* \in \mathcal{V} \backslash \mathcal{S}$ during each iteration. The basic idea is maintaining a list of images with corresponding marginal gains in descending order. Only the top image is re-evaluated during each iteration. Other images are evaluated only if the top image does not remain at the top after re-evaluation. Lazy evaluation is based on the diminishing returns property: the function value of an element cannot increase during iterations. The lazy greedy algorithm leads to a speed-up of more than 40, as we will show in the experiments.

3.4 Experiments

3.4.1 Experimental Setup

As in Chapter 2, we again evaluate our submodular reranking algorithm on the 4 public datasets: Holidays [6], UKbench [9], Oxford5k [1] and Paris6k [2], using the same features, and follow the same evaluation protocol. q in (3.6) is set to 0.9 and λ in (3.7) is set to 0.01, both fixed in all experiments. K equals the number of dataset images in each dataset; while smaller value can be used for very large datasets. $K_s = 1000$ for all datasets.

3.4.2 Comparison with Existing Approaches

As in Chapter 2, our primary focus is a retrieval algorithm that reranks database images and improves retrieval performance of multiple ranked lists obtained by multiple independent features. Although our implementation depends only on pairwise similarities without spatial verification and query expansion, the

performance by our submodular reranking is comparable to other state-of-the-art approaches using a single feature, as shown in Table 3.1. Since there are limited methods for reranking by fusion for natural image retrieval, we only compare our algorithm to [28], which is also an unsupervised reranking method using multiple features, as shown in Table 3.1. Note that [29] is not directly comparable as it requires image attributes for learning.

It is clear that our reranking algorithm outperforms [28], although we combine inferior individual features compared to [28][2]. Results by our reranking are also comparable to other state-of-the-art approaches, even we only use pairwise similarities without any learning and post-processing techniques, such as query expansion and spatial verification. We improve the best single feature (BoW) by 10.0%, 8.0%, 10.2% and 7.9% on the four datasets, respectively. Additionally, without specifically inferring weight for each feature, our reranking algorithm is very robust against inferior features, such as the color feature on Oxford5k and Paris6k, which only achieves less than 9% mAP. Although results on Oxford dataset by several approaches using a single feature [45, 49, 70] are better than those by our reranking algorithm, note that our reranking algorithm does not require SIFT descriptors or BoW vectors as [45, 49, 70] did, as long as we have pairwise similarities of pairs of images. Therefore, for the scenarios where original features cannot be stored and loaded efficiently due to limited resources, *i.e.*, mobile computing, our algorithm is more suitable than [45, 49, 70] for improving initial retrieval results. It is reasonable to expect that

[2]In [28], BoW achieved 77.5% mAP on Holidays and 3.54 N-S on UKbench, while color achieved 62.6% and 3.17, respectively. N-S score by GIST is 2.21 on UKbench.

Table 3.1: Comparisons with state-of-the-art approaches. We use N-S score on UKbench, and mAP (in %) on other datasets. "-" means the results are not reported. B, SV, MA, QE and WGC stand for baseline (single feature), spatial verification [1], multiple assignment [2], query expansion [3–5] and weakly geometric consistency [6]. Results using individual terms of our objective function are shown in the last two rows.

	Methods	Holidays	UKbench	Oxford5k	Paris6k
Baseline	BOW [45]	77.2	3.50	67.4	69.3
	VLAD [15]	55.9	3.22	32.6	38.0
	GIST [47]	35.0	1.96	24.2	19.2
	Color	55.8	3.09	8.5	8.4
B+SV/MA/QE/WGC	Philbin *et al.* [1]	-	3.45	66.4	-
	Jégou *et al.* [6]	75.1	-	54.7	-
	Jégou *et al.* [24]	84.8	3.64	68.5	-
	Wang *et al.* [10]	78.0	3.56	-	-
	Shen *et al.* [70]	76.2	3.52	75.2	74.1
	Qin *et al.* [49]	82.1	-	**78.0**	73.6
	Jégou *et al.* [14]	61.4	3.36	41.3	-
Fusion	Ours	**84.9**	**3.78**	74.3	**74.8**
	Zhang *et al.* [28]	84.6	3.77	-	-
	Zhang *et al.* [30]	80.9	3.60	68.7	-
	IG	83.9	3.75	68.5	64.6
	RRC	73.1	3.54	33.0	39.2

a higher accuracy might be obtained if we apply our reranking algorithm to fuse features which achieve better individual performance.

3.5 Discussion and Analysis

In this section, we show experimental results on how the performance changes with respect to each individual component and parameter variance.

3.5.1 Contribution of Individual Components

Our objective function consists of two terms: information gain and relative ranking consistency. These are complementary: the information gain term explores relationships between images and features at a fine level by using pairwise similarities, while the relative ranking consistency term exploits the inter-relationships between initial ranked lists in a coarser level as it only uses the ranks themselves. As shown in Table 2.1, by combining the two terms, our algorithm outperforms each individual term and achieves the best accuracy. In addition, it is reasonable that the performance by information gain term is better than that by relative ranking consistency term, since pairwise image similarities, which are continuous values, provide finer details than discrete ranks. Nevertheless, only using information gain does not produce good results on all the datasets, especially on Oxford5k and Paris6k. This reveals that rank information is complementary to information gain in matching images with significant viewpoint change.

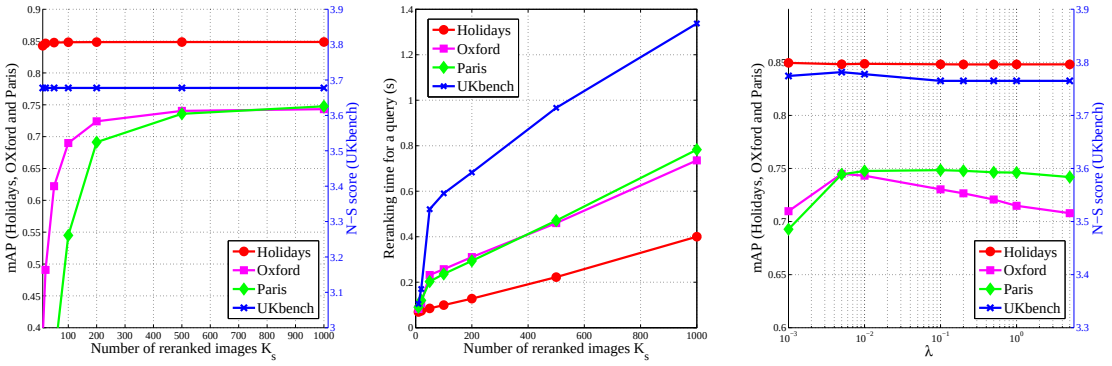
Table 3.2: Comparison of results by our reranking algorithm and other rank aggregation approaches. Runtime (in second) of reranking 1000 images for a single query using direct greedy optimization and lazy evaluation is shown in the right-most columns.

Datasets	Mean [71]	Median [72]	Geo-mean [72]	Robust [73]	Ours
Holidays	59.2	71.7	76.4	71.5	**84.9**
UKbench	2.89	3.47	3.50	3.33	**3.78**
Oxford5k	18.6	34.7	40.5	35.6	**74.3**
Paris6k	24.4	38.5	46.6	39.8	**74.8**

3.5.2 Comparison with Other Reranking Algorithms

We also compare the reranking accuracy of our reranking algorithm with other rank aggregation baseline approaches that combine multiple ranked lists. We use 5 rank aggregation approaches for comparison: mean rank aggregation [71], median rank aggregation [72], geometric mean rank aggregation [72] and robust rank aggregation [73]. The results are shown in Table 3.2.

Our reranking algorithm outperforms all other rank aggregation approaches that do not as effectively use the inter-relationships amongst multiple ranked lists. The results by mean rank aggregation are even much worse than those by a single feature (BoW), showing that a higher rank is overly diluted by other lower ranks. Incorporating the information gain and relative ranking consistency, our algorithm effectively exploits relationships of image pairs and multiple ranked lists at both a fine and a coarse level, leading to a higher retrieval accuracy.



(a) (b) (c)

Figure 3.4: (a) Change of mAP with respect to K_s. (b) Average reranking time for a single query with respect to K_s. (c) Change of mAP with respect to λ. Best view in color.

3.5.3 Parameter Evaluation

The parameter K_s controls the number of images to be reranked, which affects efficiency and reranking accuracy. Smaller K_s leads to fast convergence but may not discover images similar to queries but lower ranked since it discards a large number of initially retrieved images. We investigate the accuracy and execution time of our reranking with respect to K_s.

The retrieval accuracy in terms of mAP and average reranking time for a single query as K_s is varied are shown in Figure 3.4(a), where K_s ranges from 10 to 1000. As we perform reranking on more images, the chance of discovering a similar but lower ranked image increases. Therefore, the mAP gradually improves. More specifically, the mAP rapidly increases as K_s increases from 10 to 500 for Oxford5k and Paris6k datasets. When more images are included in reranking after this point, the improvement of mAP is only incremental, showing that reranking images

that are significantly lower ranked does not much benefit retrieval performance. In comparison, the mAP for Holidays and UKbench datasets reaches its highest value when $K_s < 100$ and remains almost constant thereafter. Images in the Oxford5k and Paris6k datasets have significant variance and each query has a large number of similar dataset images that can be retrieved. Images similar to the query can only be better discovered by a deeper inspection of initial ranked lists. In contrast, similar images in the Holidays and UKbench datasets are near-duplicates, and most queries have fewer than 10 similar images that are already highly ranked in the initial ranked lists. Therefore, only a smaller number of initially retrieved images need to be reranked.

To evaluate execution time, we calculate the average time spent to rerank K_s retrieved images for a single query in each dataset. From Figure 3.4(b), it is not surprising that reranking a larger number of images takes more time. Nevertheless, our algorithm achieves sublinear time to rerank retrieved images for a single query with respect to K_s, showing the efficiency of the greedy algorithm with lazy evaluation. Furthermore, it takes the lazy evaluation less than 1.5 seconds on a desktop with 3.4GHz CPU to rerank as many as 1000 images without any code optimization. Therefore, our reranking algorithm is scalable for large-scale image reranking tasks.

In (3.7), we balance the information gain and relative ranking consistency by parameter λ. Since λ controls the importance of individual terms, it also affects the reranking accuracy. We investigate the change of reranking performance with respect to λ, as shown in Figure 3.4(c). Our reranking algorithm is very robust: changing λ within a wide range does not affect the mAP too much, therefore we

Table 3.3: Average reranking time (in second) for a single query by direct optimization and lazy evaluation.

	Holidays	UKbench	Oxford5k	Paris6k
direct	9.46	67.63	38.33	47.12
lazy	0.23	1.62	0.73	0.85
speed-up	41×	42×	53×	55×

do not need to specifically tune λ to obtain good results. The change of mAP with respect to different λ is at most 5-6%.

3.5.4 Time Analysis

As stated in Section 3.3.4, we adopt a lazy evaluation approach to accelerate the optimization process. To show its effectiveness, we compare the reranking time for a single query by direct greedy optimization and lazy evaluation on the same machine, as shown in Table 3.3.

On all datasets, the lazy evaluation achieves more than a 40-fold speed-up compared to direct optimization. On the Oxford5k and Paris6k datasets, the lazy evaluation achieves more than a 50-fold speed-up. Therefore, our submodular reranking algorithm is very efficient and scalable for larger-scale reranking problems. With proper code optimization and parallel computing, our algorithm can be easily applied to reranking multiple ranked lists for real-time search engines.

3.6 Summary

In this chapter, we have addressed the problem of reranking images that are initially ranked by multiple features by maximizing a submodular and monotone objective function. Our objective function is composed of an information gain term and a relative ranking consistency term. The information gain term utilizes relationships of initially retrieved images based on a random walk model on a graph. Based on this term, an image initially lower ranked but resembling other retrieved images that are similar to the query will have higher rank after reranking. The relative ranking consistency term measures the relative ranking between two initially retrieved images across multiple ranked lists. It maintains the consistency of relative ranks between two images during reranking, and also captures a high rank of an image that is similar to the query but only discovered by one or a few features. The objective function can be efficiently maximized by a lazy greedy algorithm, leading to an ordered subset of initially retrieved images. Experiments show that our reranking algorithm improves overall retrieval accuracy and is computationally efficient.

Chapter 4: Multi-task Learning with Attribute Embedding for Person Re-identification

4.1 Background

In previous chapters, we have proposed two approaches to combine multiple features for generic image retrieval, where objects in images are not limited to specific categories. In this chapter, we focus on a more well-defined problem, person re-identification, which can be considered as a special application of generic image retrieval. The aim of person re-identification is to identify a person in a probe image/video by searching for the most similar instances from a gallery set. Here probe and gallery in person re-identification scenario are the same as query and database in image retrieval, respectively. The person re-identification problem is different from generic image retrieval in that: 1) database images only contain the full body of different persons that are taken by multiple non-overlapping cameras, 2) database images are well-labeled with persons' identities, and 3) the person in a probe image is guaranteed to be included in the gallery set. Due to such differences, traditional image retrieval algorithms are usually not directly applicable to person re-identification tasks. In addition, it is non-trivial to design an effective

re-identification algorithm due to large appearance, pose and illumination change across images from different cameras.

Nevertheless, even though the appearance of a person greatly changes, high level semantic concepts with respect to the person are relatively stable and consistent across different cameras. Such semantic concepts, referred to as attributes, have been widely applied to various vision applications, such as image classification and object detection, and shown promising results. When we describe an image or object by attributes, we obtain a vector in which each dimension indicates whether the corresponding attribute is present or not (or, more generally, its likelihood). In addition, it is intuitive that some attributes frequently co-occur, leading to a few subsets which contain related attributes while are mutually independent. For example, the attribute *female* is likely to be highly related to the attribute *long hair* rather than *short hair*. We show that by utilizing correlations of attributes, attributes of the same person from different cameras can be embedded into a low rank space, where embedded attributes are more accurate and informative for matching. Through the low rank attribute space, we can better match samples of the same person from one camera to another. Additionally, using this low rank embedding, we can prune noisy attributes and recover missing attributes that are introduced by inaccurate human annotation.

However, it is computationally expensive to infer attribute correlations using each pair of cameras, which also ignores the relationship of more than two cameras. To utilize relationships of features and attributes more efficiently for matching instances across cameras, we employ the Multi-Task Learning (MTL) [74] algorithm,

where one jointly learns solutions to multiple related tasks which benefit each other. MTL has been shown successful in discovering latent relationships among tasks, which cannot be found by learning each task independently. It has been widely applied to machine learning [75, 76] and computer vision [77, 78]. In addition, MTL is particularly suitable for the situation in which only a limited amount of training data is available for each task. By considering re-identifications from multiple cameras as tasks, the MTL framework can be naturally adapted to exploit features and attributes shared across cameras by learning from multiple cameras simultaneously.

4.2 Related Work

4.2.1 Person Re-identification

Person re-identification is an important research topic for video surveillance. Feature design and distance measure are two key components in solving this problem. As for feature design, different kinds of features have been tailored and employed in previous work, including histogram features from various color and textture channels [79, 80], symmetry-driven accumulation of local features [81], features from body parts with pictorial structures [82] to estimate human body configuration, and space-time features from person tracklets [83], *etc.* To use multiple features, Gray *et al.* [79] selected a subset of features by boosting for matching pedestrian images, while Liu *et al.* [84] learned person-specific weights to fuse multiple features to improve the description power of multiple features.

Considering distance measures, some works focus on learning an optimal dis-

tance metric to measure the similarity between images from two cameras. Pairwise Constrained Component Analysis [85] and Relaxed Pairwise Metric Learning [86] learn a projection from high-dimensional input space to a low-dimensional space, where the distance between pairs of data points satisfies pre-defined constraints. The Locally-Adaptive Decision Function in [87] jointly learns a distance metric and a locally adaptive thresholding rule. A Probabilistic Relative Distance Comparison model [88] attempts to maximize the likelihood of a true match which has a relatively smaller distance than a false match. A statistical inference perspective is applied in [89] to address the metric learning problem. Kernel-based distance learning has also been used [90] to handle linearly non-separable data. More recently, Zhao *et al.* [91] proposed learning mid-level filters, which mainly focuses on cross-view invariance and considers geometric configurations of body parts through patch matching. A deep learning framework to learn filter pairs that encode photometric transforms is presented in [92]. There are also approaches investigating a large camera network with more than two cameras for re-identification [93–96].

4.2.2 Attributes

Attributes are semantic concepts of objects, which are manually defined or directly learned from low level features. Previous work has investigated the correlations of attributes to improve the performance of zero/one-shot learning for attribute-based classification [97–102]. For person re-identification, attributes are powerful in preserving consistent representations of the same person and capturing

differences among different people [103–106]. However, attributes are mostly used as additional information in conjunction with low level features without considering their correlations. Although a few approaches to object classification have modeled attribute correlations [107–109], to the best of our knowledge, no work has utilized both low level features and attribute correlations across cameras for re-identification in a principled way.

4.2.3 Multi-Task Learning

Multi-Task Learning has been extensively studied. Representative work includes clustered MTL [110], Robust MTL [111] and trace norm regularization [112]. To model the shared information across tasks, a shared low rank structure is widely assumed [113, 114]. Kernel method has also been utilized to deal with linearly non-separable features [115, 116]. Dictionary learning [117] and tree sparsity constraint [118] are also incorporated into standard MTL framework. Chen *et al.* [119] applied MTL to jointly learn attribute correlations and ranking functions for image ranking. Hwang *et al.* [120] considered attribute classifiers as auxiliary tasks to object classifiers and adopted MTL to learn a shared structure for better classification and attribute prediction. Both [119] and [120] assumed attributes are related tasks while we regard cameras as tasks and infer attribute correlations by low rank embedding. For person re-identification, the multi-task support vector ranking adopted in [121] ranks individuals by transferring information of matched/unmatched image pairs from source domain to target domain. Ma *et al.* [122] also applied multi-task

learning to replace the universal distance metric for all cameras by multiple Mahalanobis distance metrics, which are different, but related, for camera pairs. We note that our approach is fundamentally different from [121] in that we explicitly model attribute correlations shared by multiple cameras, as well as low level features, without using image pairs. In addition, we seek a shared structure in terms of both low level features and attributes across multiple cameras rather than learning a metric for each pair of cameras, which can be computationally expensive.

4.3 Proposed Approach

4.3.1 Overview

In this section, we will present a **M**ulti-**T**ask **L**earning algorithm with **LO**w **R**ank **A**ttribute **E**mbedding (MTL-LORAE) for person re-identification. We aim to discover shared information amongst cameras that are treated as related tasks. Given images of people from multiple cameras, we learn a discriminative model using MTL, so that the relationships among images from these cameras can be utilized to improve the quality of the learned model. Both low level features and attributes are used in our MTL objective function. Our low rank attribute embedding is included into the objective function as well to discover relationships of attributes from multiple cameras jointly. In the embedded space, attributes of the same person from different cameras become closer, while attributes of different people become more distinct. Inaccurate and incomplete attributes can be rectified and recovered as well. The low rank structure of the embedding ensures that only a small number

of "latent" attributes contribute to the classification. We present an efficient alternating optimization method to solve the MTL-LORAE objective function. We evaluate MTL-LORAE on four person re-identification datasets and demonstrate that MTL-LORAE produces promising results.

4.3.2 Problem Formulation

We first formulate re-identification as a classification problem by learning a set of classifiers using images from multiple cameras, where a classifier corresponds to a specific person. Each gallery and probe image is then represented by a vector composed of outputs of these classifiers. By computing distance between vectors of probe and gallery images, we find and rank gallery images to complete re-identification. For simplicity, we do not distinguish between cameras and tasks, and use them interchangeably.

We are given L learning tasks $\{\mathcal{T}^1, \mathcal{T}^2, ..., \mathcal{T}^L\}$ sharing the same feature space. Our goal is to learn multi-class classifiers on a specific task using information from all tasks. In a typical multi-class setting, all tasks have the same set of C classes (persons). In a supervised one-vs-all manner, for the l-th task $\mathcal{T}^l$, we start from binary classification by considering images belonging to the c-th class as positive samples and images from all the other classes in this task as negative samples, where there are totally n_l labeled training samples. By simultaneously learning multiple tasks, our method is able to effectively transfer information from one task to another task, which is particularly desirable when training data from a task is limited. In the

following, we omit the class index c from all notation for clarity. For each training sample from the l-th task $\mathcal{T}^l$, we have a low level feature vector $\mathbf{x}_i^l \in \mathbb{R}^d$ and a label $y_i^l \in \{-1, 1\}$, where 1 indicates this sample is from the c-th class and -1 otherwise. In addition, each sample has a binary attribute vector $\mathbf{a}_i^l \in \{0, 1\}^k$, which may be semantic and labeled by humans or correspond to learned binary codes such as [123]. For each dimension of $\mathbf{a}_i^l$, 1 denotes that the corresponding attribute is present and 0 otherwise. A predictor f_l with respect to the task $\mathcal{T}^l$ will then be learned.

We can improve the discriminative and generalization ability of predictors by exploiting the relationship amongst tasks. In this way, information from task $\mathcal{T}^i$ is transferred to some other task $\mathcal{T}^j$, where training samples may be limited, so that learning the predictor f_j will benefit from learning on both $\mathcal{T}^i$ and $\mathcal{T}^j$ simultaneously. This motivates us to adopt MTL to address the problem of matching images from different cameras. In the subsequent sections, we will first introduce the low rank attribute embedding (LORAE), followed by the complete MTL formulation, the optimization algorithm and re-identification process.

4.3.3 Low Rank Attribute Embedding

A simple approach to combine low level features and attributes is to concatenate the feature vectors and original attribute vectors. However, attributes are usually inaccurate or incomplete due to the difficulty of obtaining exhaustive semantic concepts and possible inconsistency between human annotators. The absence of an attribute for an instance does not necessarily indicate that the instance

does not have that attribute, which could be incorrectly interpreted by the learning algorithm. Similarly, the presence of an attribute may be noise due to incorrect annotation. Therefore, the learned model based on the original attributes may not describe the instance accurately. Since there are a large number of attributes, they are typically related, which means some attributes often co-occur across different tasks. In this way, the presence of an attribute implies the presence of other attributes that are closely related, which helps to recover missing attributes. On the other hand, some attributes are highly independent, so that they do not occur simultaneously, which helps to remove noisy attributes.

Following [124], we learn a low rank attribute space to embed the original binary attributes into continuous attributes using attribute dependencies. In particular, there exists a transformation matrix $\mathbf{Z}$ in the low rank space converting an original attribute vector into a new vector with continuous values. The transformation matrix should capture correlations between all attributes pairs since an attribute can be affected by multiple pairs of other attributes globally. Moreover, groups of attributes can be independent from each other, suggesting the low rank property of the transformation matrix. The refined attributes capture relationships of related attributes and preserve more accurate information.

Formally, given an attribute vector $\mathbf{a}_i^l$ from task $\mathcal{T}^l$, the linear embedding is parameterized as

$$\phi_{\mathbf{Z}}(\mathbf{a}_i^l) = \mathbf{Z}^\top \mathbf{a}_i^l \quad \text{s.t.} \quad \text{rank}(\mathbf{Z}) \leq r, \tag{4.1}$$

where $\mathbf{Z} \in \mathbb{R}^{k \times k}$ is the transformation matrix, and $\text{rank}(\mathbf{Z})$ is the rank of $\mathbf{Z}$. We use

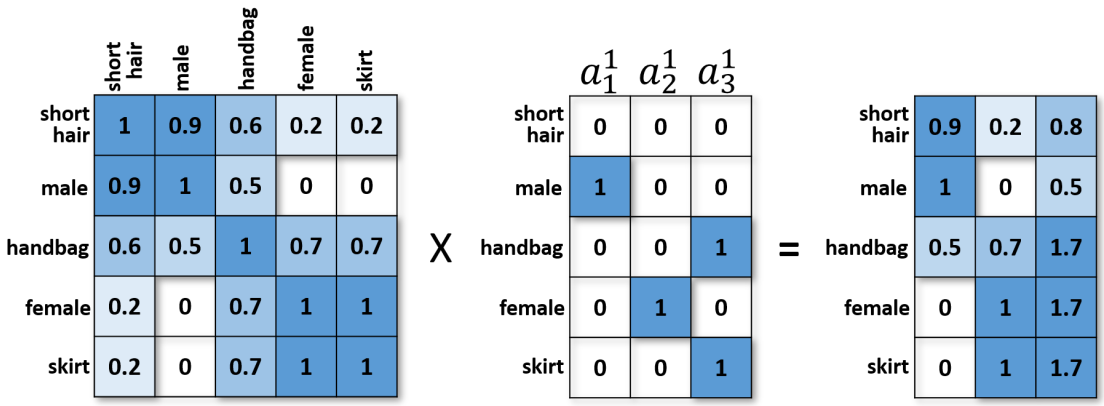


Figure 4.1: Illustration of low rank attribute embedding with three attribute vectors from task $\mathcal{T}_1$ as examples. With the learned transformation matrix, the original binary attributes are converted to continuous attributes. Semantically related attributes are recovered even though they are absent in the original attribute vectors, *i.e.*, the attribute *female* is non-zero in the embedded attribute vector due to the presence of both *skirt* and *handbag*, even though its value is 0 in the original attribute vector a_3^1.

linear embeddings although kernel methods can also be applied. The rank constraint imposed on $\mathbf{Z}$ ensures that $\mathbf{Z}$ is low rank, which means there exists a row $\mathbf{Z}_{i,:}$ (or a column $\mathbf{Z}_{:,i}$) that is a linear combination of other rows (or columns). Therefore, the parameters required for a good embedding are fewer than $k \times k$, which reduces the computational complexity. In this way, we obtain a refined attribute vector with continuous values, which better describes attribute correlations with missing values recovered and noise reduced. An intuitive illustration of the low rank embedding is presented in Figure 4.1, where missing values are successfully recovered in the embedded continuous attributes.

4.3.4 Multi-Task Learning with Low Rank Attribute Embedding

The goal of MTL is to learn task-specific predictors simultaneously using the correlations among tasks, so that the shared information can be transferred among tasks. To obtain an accurate transformation matrix $\mathbf{Z}$ for attribute embedding, we propose a unified MTL framework that utilizes attribute correlations across multiple tasks, as well as training task-specific predictors at the same time. For simplicity, we assume a linear classifier for each learning task $\mathcal{T}^l$ represented by a weight vector $\mathbf{w}^l$. For notational convenience, we concatenate the embedded attribute vector $\phi_{\mathbf{Z}}(\mathbf{a}_i^l)$ with $\mathbf{x}_i^l$ to form a new vector $\widetilde{\mathbf{x}}_i^l = [\mathbf{x}_i^l; \phi_{\mathbf{Z}}(\mathbf{a}_i^l)] \in \mathbb{R}^{d+k}$. Therefore, we have $\mathbf{w}^l \in \mathbb{R}^{d+k}$. We define the loss function as $\ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z})$ which can be any smooth and convex function measuring the discrepancy between groundtruth and predictions from learning. Specifically, we define the loss function as

$$\ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z}) = \frac{1}{2}(||y_i^l - \mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l||^2 + \gamma||\mathbf{a}_i^l - \mathbf{Z}^\top \mathbf{a}_i^l||^2). \tag{4.2}$$

The first term $||y_i^l - \mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l||^2$ is the quadratic loss from applying the learned weight vector $\mathbf{w}^l$ to the newly constructed sample $\widetilde{\mathbf{x}}_i^l$. The second term $||\mathbf{a}_i^l - \mathbf{Z}^\top \mathbf{a}_i^l||^2$ is the attribute embedding error, which regularizes the difference between original attributes and refined attributes obtained from the linear embedding through $\mathbf{Z}$. The results from the embedding should not deviate from the original attributes too much. γ controls the contributions of the two terms.

We denote all the task-specific $\mathbf{w}^l$ as a single weight matrix $\mathbf{W} = [\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^L] \in$

$\mathbb{R}^{(d+k)\times L}$. Since tasks have shared information and each task also has specific structure, similar to [114], we assume $\mathbf{W}$ is composed of a low rank matrix shared by all tasks and a task-specific sparse component representing the incoherence introduced by individual tasks. Formally, $\mathbf{W}$ can be decomposed into a low rank matrix $\mathbf{R} \in \mathbb{R}^{(d+k)\times L}$ and a sparse component $\mathbf{S} \in \mathbb{R}^{(d+k)\times L}$. Therefore, we have $\mathbf{W} = \mathbf{R}+\mathbf{S}$. Intuitively, non-zeros entries in $\mathbf{S}$ indicate the task-specific incoherence between the task and the shared low rank structure. The formulation of MTL-LORAE is then given by

$$\begin{aligned} \min_{\mathbf{R},\mathbf{S},\mathbf{Z}} \quad & \textstyle\sum_{l=1}^{L} \sum_{i=1}^{n_l} \ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z}) + \lambda ||\mathbf{S}||_0 \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{R} + \mathbf{S}, \text{rank}(\mathbf{R}) \leq r_1, \text{rank}(\mathbf{Z}) \leq r_2, \end{aligned} \tag{4.3}$$

where λ is a trade-off parameter controlling the importance of the regularization. r_1 and r_2 constrain the matrices $\mathbf{R}$ and $\mathbf{Z}$ to be low rank. $||\mathbf{S}||_0$ is the ℓ_0-norm of $\mathbf{S}$, which counts the number of non-zero entries of $\mathbf{S}$.

Solving Problem (4.3) is NP-hard since it is non-convex and non-smooth due to the sparse regularization and low rank constraints. It can be converted into a computationally tractable one by convex relaxation. First, since the ℓ_1-norm is a convex envelop of ℓ_0-norm, $||\mathbf{S}||_0$ is replaced by $||\mathbf{S}||_1$, which is the sum of all non-zero values. Second, the standard convex relaxation for the matrix rank is to use the nuclear norm (trace norm) $||\cdot||_* = \sum_i \sigma_i$, which is the sum of the singular values of a matrix. We then obtain

$$\begin{aligned} \min_{\mathbf{R},\mathbf{S},\mathbf{Z}} \quad & \textstyle\sum_{l=1}^{L} \sum_{i=1}^{n_l} \ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z}) + \lambda ||\mathbf{S}||_1 \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{R} + \mathbf{S}, ||\mathbf{R}||_* \leq r_1, ||\mathbf{Z}||_* \leq r_2, \end{aligned} \tag{4.4}$$

which is our complete MTL-LORAE formulation. For notational convenience, we denote the value of the objective function as F. By minimizing (4.4), we obtain the desired weight matrix $\mathbf{W}$ and transformation matrix $\mathbf{Z}$.

4.3.5 Optimization

The optimization of Problem (4.4) is difficult because $\mathbf{W}$ (*i.e.*, $\mathbf{R}$ and $\mathbf{S}$) and $\mathbf{Z}$ are coupled together by $\widetilde{\mathbf{x}}_i^l$. However, by alternating between optimizing the objective function with respect to one variable and fixing the other one, the problem is solvable. When fixing $\mathbf{Z}$, $||\mathbf{a}_i^l - \mathbf{Z}^\top \mathbf{a}_i^l||^2$ becomes a constant so it can be omitted. $\widetilde{\mathbf{x}}_i^l$ is also constant with respect to $\mathbf{w}^l$, so that it can be regarded as an ordinary training sample. By removing the nuclear norm constraint on $\mathbf{Z}$, Problem (4.4) reduces to the standard MTL formulation under the assumption of shared low rank structure plus incoherent sparse values

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \sum_{l=1}^{L}\sum_{i=1}^{n_l} \ell'(y_i^l, \widetilde{\mathbf{x}}_i^l) + \lambda||\mathbf{S}||_1 \\ \text{s.t.} & \mathbf{W} = \mathbf{R} + \mathbf{S}, ||\mathbf{R}||_* \leq r_1 \end{array}, \tag{4.5}$$

where $\ell'(y_i^l, \widetilde{\mathbf{x}}_i^l) = \frac{1}{2}||y_i^l - \mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l||^2$. Problem (4.5) can be solved by the *MixedNorm* approach from [114]. Details can be found in [114].

When fixing $\mathbf{W}$, both $\mathbf{R}$ and $\mathbf{S}$ become constant, so we can remove the constraints related to them. Therefore, we obtain the objective function

$$\begin{array}{ll} \min\limits_{\mathbf{Z}} & \sum_{l=1}^{L}\sum_{i=1}^{n_l} \ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z}) \\ \text{s.t.} & ||\mathbf{Z}||_* \leq r_2 \end{array}. \tag{4.6}$$

Relaxing the constraint as a regularization term, we obtain

$$\min_{\mathbf{Z}} \quad \sum_{l=1}^{L} \sum_{i=1}^{n_l} \ell(y_i^l, \mathbf{a}_i^l, \widetilde{\mathbf{x}}_i^l, \mathbf{Z}) + \beta ||\mathbf{Z}||_* \ . \tag{4.7}$$

With the nuclear norm regularization, the optimal transformation matrix $\mathbf{Z}$ will not degenerate to a trivial solution, *i.e.*, an identity matrix $\mathbf{I}$. However, due to the non-smooth nuclear constraint on $\mathbf{Z}$, it is not easy to optimize (4.7). For clarity of notation, we denote the loss function with respect to $\mathbf{Z}$ as $\ell_{\mathbf{Z}}$, and the regularization term as $h_{\mathbf{Z}} = ||\mathbf{Z}||_*$. Problem (4.7) is then rewritten as

$$\min_{\mathbf{Z}} \quad \ell_{\mathbf{Z}} + \beta h_{\mathbf{Z}} \ . \tag{4.8}$$

$\ell_{\mathbf{Z}}$ is convex, differentiable and Lipschitz continuous. $h_{\mathbf{Z}}$ is convex but non-differentiable. Thus, (4.8) can be solved by the proximal gradient method iteratively.

First, we represent the gradient of $\ell_{\mathbf{Z}}$ with respect to $\mathbf{Z}$ as $\partial_{\mathbf{Z}}\ell$. According to the proximal gradient algorithm, at each iteration step j, we then have $\mathbf{Z}_j = \mathbf{prox}_{t_j}(\mathbf{Z}_{j-1} - t_j \partial_{\mathbf{Z}_{j-1}}\ell)$, where $t_j > 0$ is the step size and j is the iteration index. $\mathbf{prox}_{t_j}$ is a proximal operator, defined as

$$\begin{aligned} \arg\min_{\mathbf{Z}} \quad & \ell_{\mathbf{Z}_{j-1}} + \langle \partial_{\mathbf{Z}_{j-1}}\ell, \mathbf{Z} - \mathbf{Z}_{j-1} \rangle \\ & + \tfrac{1}{2t_j} ||\mathbf{Z} - \mathbf{Z}_{j-1}||_F^2 + \beta h_{\mathbf{Z}} \end{aligned} \ , \tag{4.9}$$

where $\langle \cdot, \cdot \rangle$ is the inner product. (4.9) finds the $\mathbf{Z}$ that minimizes the surrogate of the loss function ℓ at point $\mathbf{Z}_{j-1}$ plus a quadratic proximal regularization term and

the non-smooth regularization term. (4.9) can be simplified to

$$\arg\min_{\mathbf{Z}} \quad \tfrac{1}{2t_j}||\mathbf{Z}-(\mathbf{Z}_{j-1}-t_j\ell_{\mathbf{Z}_{j-1}})||_F^2+\beta h_{\mathbf{Z}} \ . \tag{4.10}$$

It is clear that (4.10) can be effectively solved by performing SVD on $\mathbf{Z}_{j-1}-t_j\ell_{\mathbf{Z}_{j-1}}$ and then soft-thresholding the singular values.

In practice, we adopt the Accelerated Gradient Method (AGM) [112] to accelerate the optimization. AGM adaptively estimates the step size and introduces the search point $\widetilde{\mathbf{Z}_j}$ that is a linear combination of the latest two approximations $\mathbf{Z}_{j-1}$ and $\mathbf{Z}_{j-2}$, $\widetilde{\mathbf{Z}}_j=\mathbf{Z}_{j-1}+(\frac{\alpha_{j-1}-1}{\alpha_j})(\mathbf{Z}_{j-1}-\mathbf{Z}_{j-2})$. Here, α_{j-1} and α_j control the combination weights of the previous two approximations, which are also updated iteratively by $\alpha_j=\frac{1+\sqrt{1+4\alpha_{j-1}^2}}{2}$ with $\alpha_0=1$. The gradient in the j-th iteration is then performed on $\widetilde{\mathbf{Z}}_j$ instead of $\mathbf{Z}_j$, where $\widetilde{\mathbf{Z}}_1=\mathbf{Z}_0$.

The gradient $\partial_{\mathbf{Z}}\ell$ is explicitly computed as

$$\begin{aligned}
\partial_{\mathbf{Z}}\ell &= (y_i^l-\mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l)\tfrac{\partial \mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l}{\partial \mathbf{Z}}+\gamma\tfrac{\partial \mathbf{Z}^\top\mathbf{a}_i^l}{\partial \mathbf{Z}}(\mathbf{a}_i^l-\mathbf{Z}^\top\mathbf{a}_i^l)^\top \\
&= (y_i^l-\mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l)\tfrac{\partial \mathbf{w}_\phi^{l\top}\mathbf{Z}^\top\mathbf{a}_i^l}{\partial \mathbf{Z}}+\gamma\tfrac{\partial \mathbf{Z}^\top\mathbf{a}_i^l}{\partial \mathbf{Z}}(\mathbf{a}_i^l-\mathbf{Z}^\top\mathbf{a}_i^l)^\top \\
&= (y_i^l-\mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l)\mathbf{a}_i^l\mathbf{w}_\phi^{l\top}+\gamma\mathbf{a}_i^l(\mathbf{a}_i^l-\mathbf{Z}^\top\mathbf{a}_i^l)^\top \\
&= \mathbf{a}_i^l[\mathbf{w}_\phi^{l\top}(y_i^l-\mathbf{w}^{l\top}\widetilde{\mathbf{x}}_i^l)+\gamma(\mathbf{a}_i^l-\mathbf{Z}^\top\mathbf{a}_i^l)^\top],
\end{aligned} \tag{4.11}$$

where $\mathbf{w}_\phi^l\in\mathbb{R}^k$ is part of the weight vector $\mathbf{w}^l$ corresponding to the embedded attribute $\phi_{\mathbf{Z}}(\mathbf{a}_i^l)$. When the optimization for $\mathbf{Z}$ converges, we update $\mathbf{Z}$, fix it and minimize the objective function for $\mathbf{W}$. The optimization will stop after a predefined iteration number P or when the difference $\Delta F=F_{j-1}-F_j>0$ between

Algorithm 3 Multi-Task Learning with Low Rank Attribute Embedding (MTL-LORAE)

Input: Training data samples $\{\mathbf{x}_i^l, \mathbf{a}_i^l, y_i^l\}$ for all L tasks, initial $\mathbf{Z}_0$ and $\mathbf{W}_0$, iteration number P and threshold $th > 0$ to control iteration step.

Output: Learned $\mathbf{Z}$ and $\mathbf{W}$.

$\mathbf{Z} \leftarrow \mathbf{Z}_0$, $\mathbf{W} \leftarrow \mathbf{W}_0$;
Evaluate objective function F_0 using $\mathbf{Z}$ and $\mathbf{W}$;
for $j = 1$ **to** P **do**
    Optimize (4.5) when fixing $\mathbf{Z}$ by *MixedNorm* [114];
    Update $\mathbf{W} \leftarrow \mathbf{W}_j$;
    Optimize (4.6) when fixing $\mathbf{W}$ by AGM algorithm [112];
    Update $\mathbf{Z} \leftarrow \mathbf{Z}_j$;
    Evaluate objective function F_j;
    Calculate $\Delta F = F_{j-1} - F_j$;
    if $\Delta F < th$ **then**
        break;
    end if
end for

consecutive values of the objective function is below a threshold. The entire optimization process is summarized in Algorithm 3.

4.3.6 Re-identification Process

With C training classes (persons), we obtain C class-specific weight matrices and transformation matrices, each of which is denoted as $\mathbf{W}_{(c)} = [\mathbf{w}^1_{(c)}, \mathbf{w}^2_{(c)}, \cdots, \mathbf{w}^L_{(c)}]$ and $\mathbf{Z}_{(c)}$, respectively, by performing the optimization with respect to each class. Given an image taken by the l'-th camera, $l' = 1, 2, \cdots, L$, which is either from the gallery or the probe set, we first extract low level feature $\mathbf{x}^{l'}$ and attribute vector $\mathbf{a}^{l'}$. By applying the transformation matrices, we convert our feature and attribute vectors to a new set of vectors, denoted as $\widetilde{\mathbf{X}}^{l'} = [\widetilde{\mathbf{x}}^{l'}_{(1)}, \widetilde{\mathbf{x}}^{l'}_{(2)}, \cdots, \widetilde{\mathbf{x}}^{l'}_{(C)}] \in \mathbb{R}^{(d+k)\times C}$, where the c-th column $\widetilde{\mathbf{x}}^{l'}_{(c)} = [\mathbf{x}^{l'}; \mathbf{Z}^{\top}_{(c)}\mathbf{a}^{l'}]$ is the concatenation of the feature vector and the embedded attribute vector using the c-th transformation matrix $\mathbf{Z}_{(c)}$. We further select weight vectors with respect to l'-th task from C weight matrices, and multiply them with the new vectors to obtain a score vector $\mathbf{s}$ as

$$\mathbf{s} = [\mathbf{w}^{l'\top}_{(1)}\widetilde{\mathbf{x}}^{l'}_{(1)}, \mathbf{w}^{l'\top}_{(2)}\widetilde{\mathbf{x}}^{l'}_{(2)}, \cdots, \mathbf{w}^{l'\top}_{(C)}\widetilde{\mathbf{x}}^{l'}_{(C)}], \tag{4.12}$$

where $\mathbf{w}^{l'}_{(c)}$ is the column weight vector extracted from $\mathbf{W}_{(c)}$ corresponding to the l'-th task $\mathcal{T}^{l'}$ trained for the c-th class. Therefore, each image is finally represented by a C-dimensional score vector $\mathbf{s}$, similar to the reference coding method in [125] and [126]. The similarity between a gallery image and a probe image is then measured by the Euclidean distance between two score vectors. Note that the classes in the training set can be the same as or disjoint from those in the gallery and probe sets.

For multi-shot scenarios, multiple images are presented for each probe/gallery. Given a probe image set containing m_p images, the re-identification process needs to aggregate image-level similarities to rank the gallery image sets. To this end, we adopt the following voting scheme. We first compute the distances between m_p probe images and all gallery images, and then apply a Gaussian kernel to convert the distances to similarities. To obtain a single similarity between the probe and a gallery image set of m_g images, we sum up all $m_p \times m_g$ similarities and divide the sum by the number of gallery images, m_g, to discount the affect of a gallery set that contains many images.

4.4 Experiments

4.4.1 Datasets

We evaluate our approach on 4 public datasets, iLIDS-VID [83], PRID [127] and VIPeR [128] and SAIVT-SoftBio [93]. The iLIDS-VID dataset consists of 600 image sets for 300 people from two cameras at an airport, which is designed for multi-shot re-identification. Each person has two image sets from the two cameras respectively, where each image set contains 23 to 192 images, sampled from a short video taken within a few seconds. The PRID dataset is used for single-shot scenario; it contains images of different people from two cameras, A and B, under different illumination and background conditions. There are 385 and 749 people appearing in cameras A and B, respectively, of which 200 appear in both cameras. The VIPeR dataset contains 632 persons from two cameras, with only one image

per person in each camera. The SAIVT-SoftBio dataset is also designed for multi-shot re-identification, where images are also extracted from a short video containing a person. There are 152 people from 8 different cameras. Since not every person appears in all cameras, following the evaluation setting in [96], we select those appearing in three cameras (#3, #5 and #8) as our evaluation set.

4.4.2 Implementation Details

We use a 2784-dimensional color and texture descriptor [79] as our low level feature representation. It is composed of 8 color channels (RGB, HSV and YCbCr [1]) and 19 texture channels (Gabor and Schmid). As for attributes, we learn binary SVMs as in [105] to predict the same 20-bit attributes in [105] for PRID and 90-bit attributes in [129] for VIPeR. For other datasets, we learn attribute functions by [130] in an unsupervised manner on the training set and generate 32-bit attributes. Following the standard evaluation protocols, we randomly select 150, 100 and 316 persons appearing in all cameras as our training set for iLIDS-VID, PRID and VIPeR, respectively, while the remaining 150, 649 and 316 persons serve as the test set (galleries and probes). All the results are averaged over 10 random training/test splits. Parameters for learning are empirically set via cross-validation and fixed for all experiments. $r_1 = 2$, $r_2 = 5$ and $\lambda = 0.3$ in (4.3). $\gamma = 0.5$ in (4.2). Iteration number $P = 500$ and threshold $th = 10^{-5}$ in Algorithm 3.

[1] Only one of the luminance channels (V and Y) is used.

4.4.3 Experimental Results

4.4.3.1 iLIDS-VID

Among 150 persons in the test set, image sets from one camera are used as the probe set, while those from another camera serve as the gallery set. We first compare our approach with 8 completing methods for multi-short re-identification: Salience Matching (Salmatch) [131], Learning Mid-level Filters (LMF) [91], Multi-short Symmetry-driven Accumulation of Local Features (MS-SDALF) [81], Multi-short color with RankSVM (MS-color+RSVM) [83], Multi-short color&LBP with RankSVM (MS-color&LBP+RSVM) [83], color&LBP with Dynamic Time Warping (Color&LBP+DTW) [86], HoGHoF with DTW (HOGHOF+DTW) [132], color&LBP with Discriminative Video fragments selection and Ranking (MS-color&LBP+DVR) [83]. We use cumulative match characteristic (CMC) curves to evaluate performance, and show experimental results in Figure 4.2 and Table 4.1.

Our MTL-LOREA approach produces the best results consistently in terms of matching rate with respect to varying ranks. Specifically, when inspecting the matching rate at rank 1 and rank 5, we find a relatively large improvement compared to the best existing method, MS-color&LBP+DVR. Specifically, our method successfully increases the rank 1 accuracy from 34.5% to 43.0%, resulting in an 8.5% improvement. In addition, we obtain nearly 100% matching rate at rank 50, while most compared methods can only achieve 80% matching rate or even less.

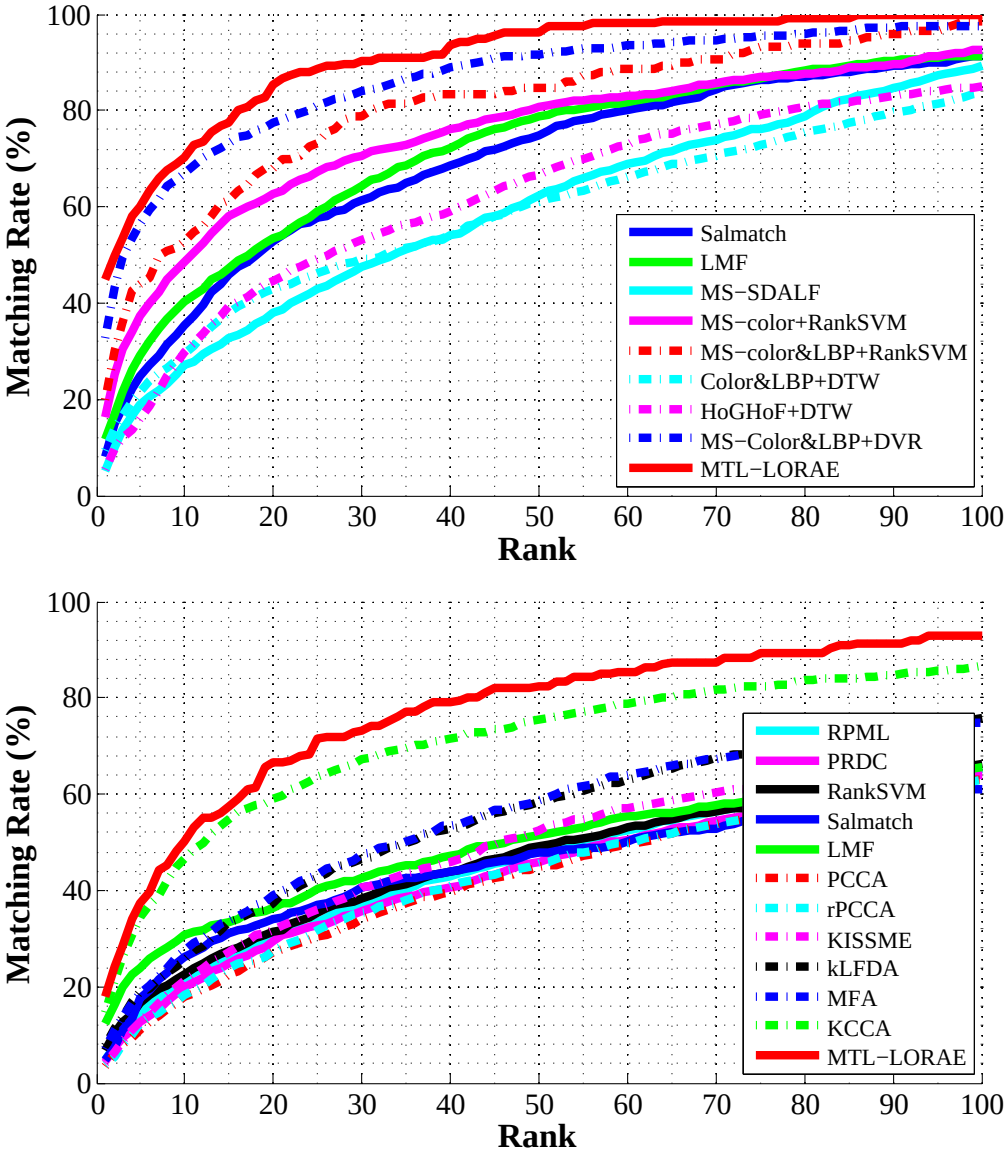


Figure 4.2: CMC curves of our approach and state-of-the-art approaches on the iLIDS-VID dataset (top) and PRID dataset (bottom).

Table 4.1: CMC scores of ranks from 1 to 50 on the iLIDS-VID dataset. Numbers indicate the percentage (%) of correct matches within a specific rank.

Rank	1	5	10	20	30	50
Salmatch [131]	8.0	24.8	35.4	52.9	61.3	74.8
LMF [91]	11.7	29.0	40.3	53.4	64.3	78.8
MS-SDALF [81]	5.1	19.0	27.1	37.9	47.5	62.4
MS-color+RSVM [83]	16.4	37.3	48.5	62.6	70.7	80.6
MS-color&LBP+RSVM [83]	20.0	44.0	52.7	68.0	78.7	84.7
Color&LBP+DTW [83]	9.3	21.6	29.5	43.0	49.1	61.0
HoGHoF+DTW [83]	5.3	16.0	29.7	44.7	53.1	66.7
MS-color&LBP+DVR [83]	34.5	56.4	67.0	77.4	84.0	91.7
MTL-LOREA	**43.0**	**60.0**	**70.2**	**85.3**	**90.2**	**96.3**

4.4.3.2 PRID

Following the protocol in [127], we use images of 100 persons from camera A as the probe set, and 649 persons in camera B as the gallery set, excluding all training samples. We compare our algorithm with 11 learning-based methods [2]: Relaxed Pairwise Metric Learning (RPML) [86], Probabilistic Relative Distance Comparison (PRDC) [88], RankSVM (RSVM) [133], Salmatch [131], LMF [91], Pairwise Constrained Component Analysis (PCCA) [85], regularized PCCA (rPCCA) [90], Keep It Simple and Straightforward MEtric (KISSME) [89], kernel Local Fisher

[2]We do not compare with DVR [83] because DVR only uses 89 persons for testing.

Table 4.2: CMC scores of ranks from 1 to 50 on the PRID dataset. Numbers indicate the percentage (%) of correct matches within a specific rank.

Rank	1	5	10	20	30	50
RPML [86]	4.8	14.3	21.6	30.2	37.2	48.1
PRDC [88]	4.5	12.6	19.7	29.5	35.8	46.0
RSVM [133]	6.8	16.5	22.7	31.5	38.4	49.3
Salmatch [131]	4.9	17.5	26.1	33.9	40.5	47.8
LMF [91]	12.5	23.9	30.7	36.5	42.6	51.6
PCCA [85]	3.5	10.9	17.9	27.1	34.2	45.0
rPCCA [90]	3.8	12.3	18.3	27.5	35.2	45.4
KISSME [89]	4.1	12.8	21.1	31.8	40.7	52.5
kLFDA [90]	7.6	18.9	25.6	37.4	46.7	58.5
MFA [90]	7.2	18.7	27.6	39.1	47.4	58.7
KCCA [134]	14.5	34.3	46.7	59.1	67.2	75.4
MTL-LOREA	**18.0**	**37.4**	**50.1**	**66.6**	**73.1**	**82.3**

Discriminant Classifier (kLFDA) [90], Marginal Fisher Analysis (MFA) [90] and Kernel Canonical Correlation Analysis (KCCA) [134]. We again use CMC curves to evaluate performance, as shown in Figure 4.2 and Table 4.2.

Our MTL-LOREA approach outperforms all existing methods by a large margin. In particular, our approach achieves 50% matching rate at rank 10, while the matching rate of most other approaches is less than 30%. Except for our approach

and KCCA, all other methods are only able to obtain a 50% matching rate as far as rank 55. Our approach also consistently outperforms KCCA, which currently holds state-of-the-art performance, from the beginning. Specifically, on average the absolute improvement in terms of matching rate by our approach over KCCA is 6%, where the margin gradually increases as we move from lower ranks to higher ranks. Notably, the relative improvement by our approach over KCCA is nearly 10%. In terms of the accuracy at rank 1 and rank 5, our approach achieves a matching rate 18% at rank 1 and 37.4% at rank 5, respectively, leading to a 3.5% and 3.1% performance gain at rank 1 and rank 5 over KCCA. When evaluated with more retrieved samples, our approach still secures the best performance. Pairwise distance metric learning based on camera pairs is clearly not powerful enough to obtain good results. Although using kernel tricks, without fully investigating the relationships of features and attributes from multiple cameras, KCCA cannot improve the performance much. The experiments further verify that MTL-LOREA, which learns attribute correlations in an MTL setting with low rank embedding, successfully exploits relationships among attributes, thus producing a more discriminative model.

Since all the competing methods only use low level features while MTL-LOREA adopts both low level features and attributes, we conduct additional experiments on the PRID dataset, where semantic attributes are provided, to verify that the performance boost by MTL-LOREA results from our learning framework rather than attributes only. We collect publicly available implementations of 5 existing approaches, which are Salmatch [131], LMF [91], rPCCA [90], kLFDA [90] and MFA [90]. We concatenate the original binary attribute vectors and low level

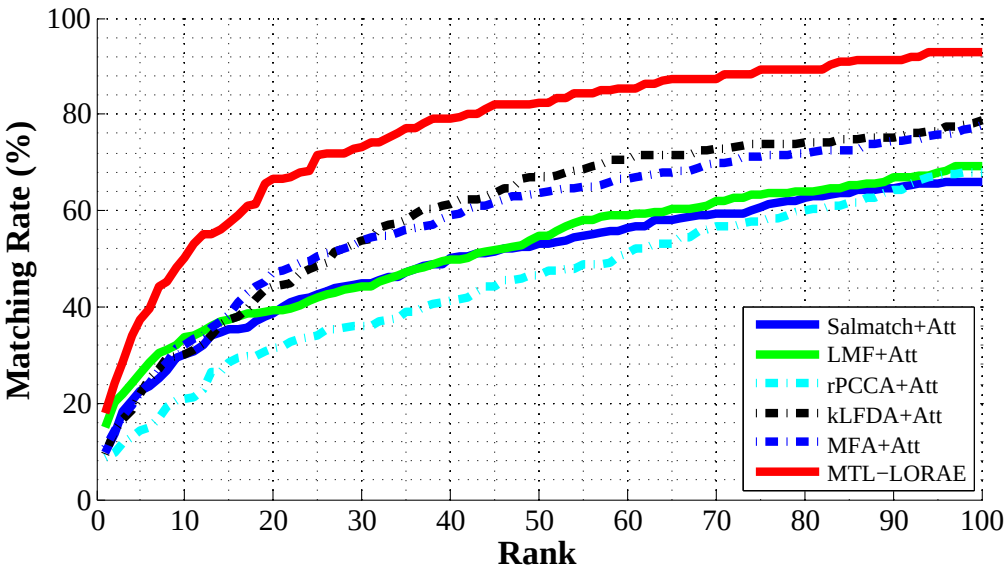


Figure 4.3: CMC curves of our approach and 5 state-of-the-art approaches with attributes added on the PRID dataset.

features used by each approach to form a set of new feature vectors, while keeping other parts of each implementation unchanged. For fair comparison, we use the default parameter setting provided by original authors for each implementation. The comparisons are shown in Figure 4.3 and Table 4.3.

With attribute added, all the 5 compared methods produce better results, justifying the use of attributes. Nevertheless, the performance of the 5 compared methods is still worse than that of our MTL-LOREA approach, which again verifies that our learning framework with MTL and low rank attribute embedding is effective in utilizing shared information amongst tasks, as well as exploiting attribute correlations, to improve the re-identification accuracy.

Table 4.3: CMC scores of our approach and 5 state-of-the-art approaches with attributes added at ranks from 1 to 50 on the PRID dataset. Numbers indicate the percentage (%) of correct matches within a specific rank. "Att" indicates attributes are added to the original features.

Rank	1	5	10	20	30	50
Salmatch [131]	4.9	17.5	26.1	33.9	40.5	47.8
Salmatch+Att	9.6	22.6	30.2	38.8	44.8	53.1
LMF [91]	12.5	23.9	30.7	36.5	42.6	51.6
LMF+Att	15.0	26.2	33.6	39.3	44.1	54.7
rPCCA [90]	3.8	12.3	18.3	27.5	35.2	45.4
rPCCA+Att	8.7	14.4	20.8	31.5	36.0	46.7
kLFDA [90]	7.6	18.9	25.6	37.4	46.7	58.5
kLFDA+Att	9.4	22.0	30.2	44.1	53.9	66.8
MFA [90]	7.2	18.7	27.6	39.1	47.4	58.7
MFA+Att	10.7	22.1	32.0	47.3	53.8	63.7
MTL-LOREA	**18.0**	**37.4**	**50.1**	**66.6**	**73.1**	**82.3**

4.4.3.3 VIPeR

Since our approach requires multiple images to learn the MTL model, we apply data augmentation to generate enough training samples for MTL-LORAE. For each training image, we apply horizontal and vertical translation $t \in \{-6, -3, 0, 3, 6\}$ pixels and clockwise rotation $r \in \{-5, 0, 5\}$ degrees, resulting in totally 75 images. We compare MTL-LORAE with 4 best-performing methods, including 2 recent ones:

Table 4.4: CMC scores of ranks from 1 to 20 on the VIPeR dataset. Numbers indicate the percentage (%) of correct matches within a specific rank.

Rank	1	5	10	20
kLFDA [90]	32.2	65.8	79.7	90.9
KCCA [134]	37.3	71.4	**84.6**	92.3
LX [135]	40.0	68.9	80.5	91.1
TSR [136]	31.6	68.6	82.8	**94.6**
MTL-LORAE	**42.3**	**72.2**	81.6	89.6

LOMO+XQDA (LX) [135] and TSR [136], as shown in Table 4.4. Our MTL-LORAE achieves the best accuracy at rank 1 and rank 5, outperforming existing methods by a large margin, and comparable results at rank 10 and rank 20.

4.4.3.4 SAIVT-SoftBio

We use half of the people as the training set and the remaining half as the test set. In the test set, each image set serves as the probe while all the remaining image sets are regarded as the gallery. For fair comparison, we evaluate the performance using precision, recall and F_1-score by regarding the identification problem as a classification problem as [96] does, instead of CMC score that is not applicable to the scenario with more than two cameras. We compare our algorithm to RSVM [133], KISSME [89], RSVM with Conditional Random Field (R-CRF) [96], and KISSME with Conditional Random Field (K-CRF) [96]. Results by our approach and other

competing methods with respect to each pair of cameras, as well as results averaged over all possible camera pairs, are presented in Table **??**. Our MTL-LOREA is able to achieve the best F_1-score, outperforming the best existing method, K-CRF, by 4.6%. In addition, MTL-LOREA achieves the second best recall rate and comparable precision rate. Without explicitly handling pairs of cameras, MTL-LOREA still successfully captures the relationship between two cameras and significantly improves the performance, which verifies our approach of exploiting shared information across cameras and further justifies the use of MTL. We also note that our learning framework can learn the models for all cameras simultaneously regardless of the number of cameras, which is more computationally efficient than existing methods that explicitly deal with all pairs of cameras. In addition to the comparisons in terms of precision, recall and F_1-score averaged over all camera pairs in our paper, we further show comparisons of our approach and other competing methods with respect to each pair of cameras separately in Table 4.5. Compared with 4 competing methods, our MTL-LOREA approach achieves better or comparable precision and recall, and the best F_1-score on all the three camera pairs, showing its outstanding capability of discovering and identifying a person accurately.

4.5 Discussions and Analysis

We conduct further experiments to better understand the characteristics of our MTL-LOREA formulation and analyze the contribution of its individual components.

Table 4.5: Comparison of precision, recall and F_1-score (in %) regarding all camera pairs by existing methods and our approach on SAIVT-SoftBio dataset. *C3*, *C5* and *C8* represent cameras #3, #5 and #8.

	RSVM [133]	KISSME [89]	R-CRF [96]	K-CRF [96]	MTL-LOREA
C3-C5					
Precision	14.9	15.9	37.2	38.0	**38.1**
Recall	24.7	50.3	15.5	28.5	**75.1**
F_1-score	15.9	23.4	18.2	30.3	**50.5**
C3-C8					
Precision	27.7	20.7	**55.4**	48.4	41.0
Recall	29.4	70.1	43.1	51.1	**65.6**
F_1-score	20.1	31.0	43.4	47.6	**50.4**
C5-C8					
Precision	25.7	19.9	45.2	**47.1**	36.8
Recall	43.4	**65.4**	30.8	44.7	53.8
F_1-score	24.6	29.6	32.4	**43.7**	**43.7**
Average					
Precision	22.0	19.7	**53.7**	50.3	45.2
Recall	42.1	**66.1**	39.4	49.8	63.7
F_1-score	26.2	29.5	42.0	48.3	**52.9**

4.5.1 Convergence Analysis

Our original formulation in (4.4) is difficult to optimize. However, by alternating between optimizing the objective function with respect to one variable and fixing the other one, we can solve this problem. When fixing $\mathbf{Z}$, we obtain Problem (4.5) as shown in the submission, which can be solved by *MixedNorm* approach in [114]. The optimization algorithm of *MixedNorm* approach [114] guarantees the global convergence with a convergence rate $\mathcal{O}(1/k^2)$, where k is the iteration number. On the other hand, when fixing $\mathbf{W}$, both the loss function $\ell_{\mathbf{Z}}$ and regularization term $h_{\mathbf{Z}}$ in Problem (4.8) are convex, so that a global optimal is available. By adopting the Accelerated Gradient Method (AGM) in [112], we can achieve a convergence rate as $\mathcal{O}(1/k^2)$. Proofs with respect to the convergence rate can be found in [112], [114] and [137]. Therefore, our approach will find the global optimal via alternating optimization.

To investigate the convergence performance of MTL-LOREA, we visualize the change of objective function value during the optimization in Figure 4.4. The optimization is conducted regarding a randomly selected person from the training set on the iLIDS-VID and PRID datasets, respectively. The objective function value quickly decreases and reaches its minimal after a few iterations, verifying the effectiveness of our optimization strategy.

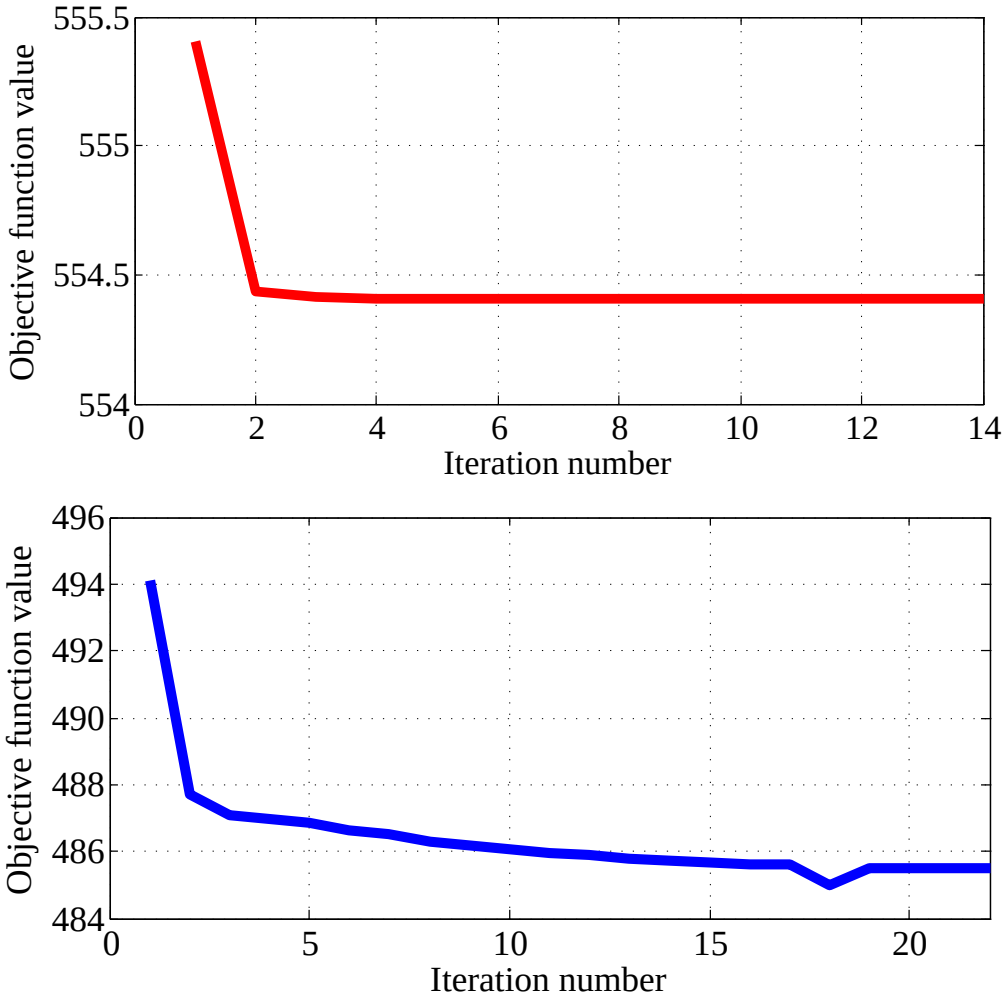


Figure 4.4: Change of objective function value during optimization on the iLIDS-VID dataset (top) and PRID dataset (bottom).

4.5.2 Analysis on the Transformation Matrix

Based on the assumption that attributes are usually correlated, the learned low rank matrix $\mathbf{Z}$ should preserve attribution correlations well. In Figure 4.5, we show the full learned transformation matrix $\mathbf{Z}$ averaged over 100 people from the training set on *PRID* since the attributes are manually defined and have semantic meaning. Since the data-driven attributes learned by [130] do not preserve clear semantic meaning, we do not show the learned transformation matrix here. Clearly, some attributes are closely related so that they have higher correlation score, *i.e.*, the attributes *shorts* and *barelegs*, since they should frequently co-occur. In contrast, a person cannot wear *light bottoms* (or *light shirt*) and *dark bottoms* (or *dark shirt*) at the same time so that these two attributes have negative correlation. As another example, the attribute *skirt* has positive correlation with the attribute *barelegs*, while it has negative correlation with the attribute *male*. Similarly, it is also reasonable that the attribute *hassatchel* has negative correlation with both the attributes *hashandbagcarrierbag* and *hasbackpack* since a person is unlikely to carry different bags simultaneously. The learned transformation matrix captures the correlations amongst attributes well and thus improves the quality of the original attributes, which justifies the effectiveness of the low rank structure of the embedding space and our learning framework.

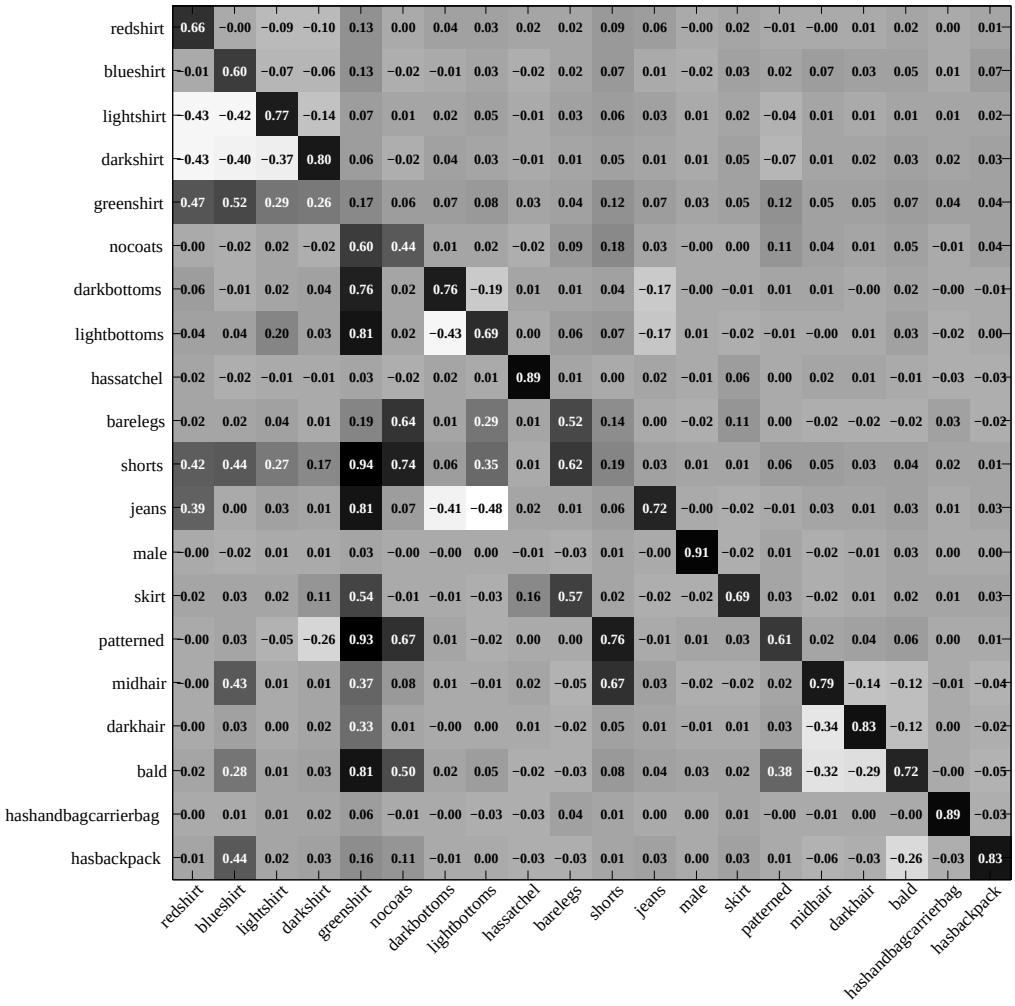


Figure 4.5: Attribute correlations learned on the PRID dataset. Larger values indicate two attribute are more positively correlated.

4.5.3 Evaluation of Individual Components

To verify the effect of individual components in our framework and show that each of them contributes to the performance boost, we evaluate three variants of our approach. Instead of MTL, we assume tasks are independent and learn classifiers for each task separately while keeping other components unchanged, so that the learning is based on single tasks (STL). We also use the original attributes without

embedding, and discard the embedding error term in the objective function in (4.2) to have another variant, MTL-Att. In addition, we remove the low rank constraint on $\mathbf{Z}$ in (4.4), which embeds original attributes to a possible full rank space by making attributes highly uncorrelated. We denote this variant as MTL-FR. We then evaluate the three variants on iLIDS-VID and PRID to see how each component affects the performance.

We show CMC scores at some ranks in Table 4.6 and display the CMC curves in Figure 4.6. The results by STL are always worse than those by MTL-LOREA and other two MTL-based variants, which indicates that learning related tasks simultaneously successfully exploits shared information amongst tasks and thus increases the discriminative ability of the learned model. We also find that MTL-FR is inferior to MTL-Att, suggesting that assuming attributes are uncorrelated is unreasonable and even hurts performance. However, only using the original attributes without investigating their correlations, MTL-Att cannot produce the best results, although it already outperforms most existing approaches. The experiments reveal that individual components, *i.e.*, MTL and low rank embedding, are integrated into our formulation in a principled way and together improve the performance.

4.6 Summary

In this chapter, we have proposed a multi-task learning (MTL) formulation with low rank attribute embedding for person re-identification. Multiple cameras are treated as related tasks, whose relationships are decomposed as a low rank struc-

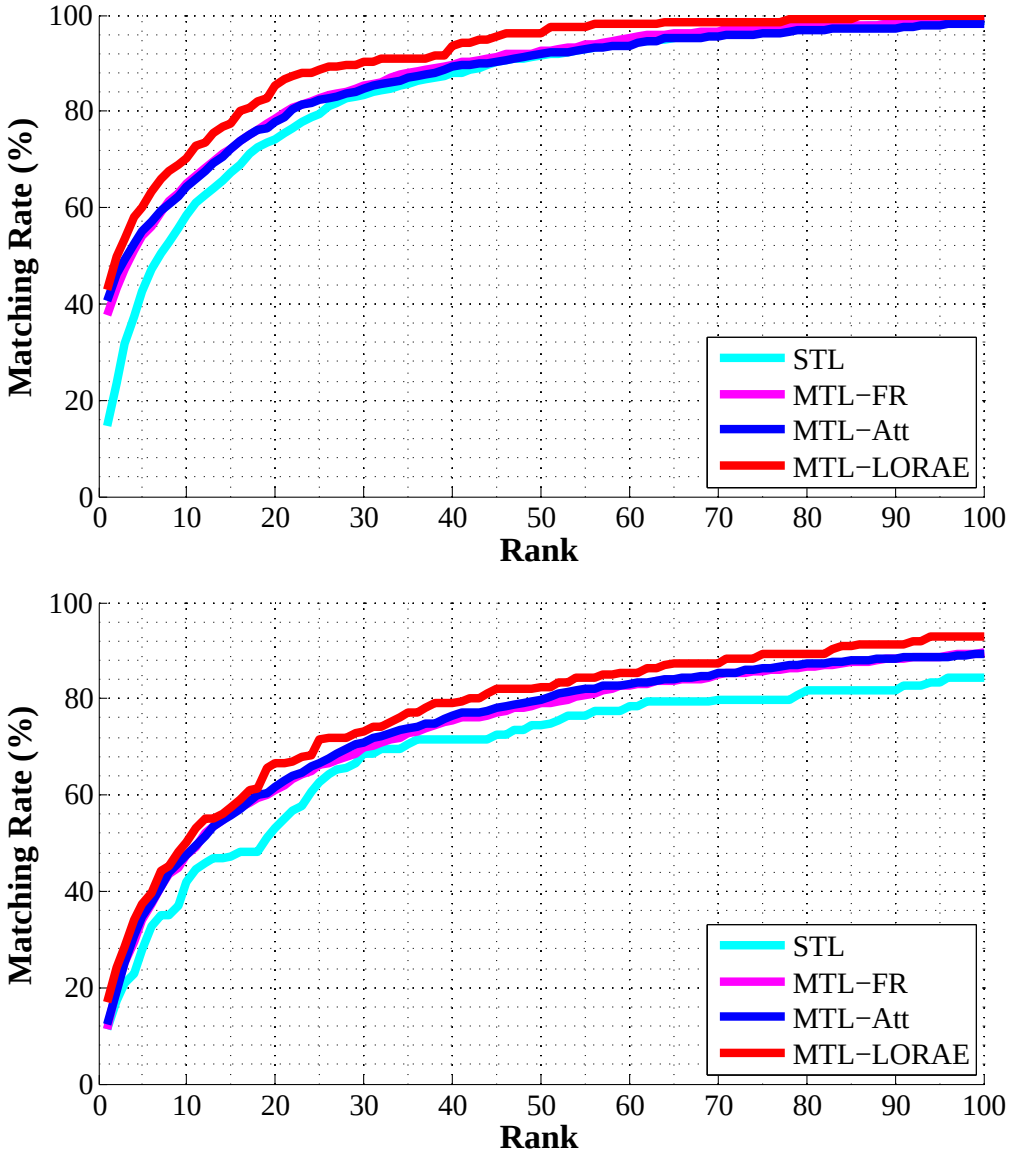


Figure 4.6: CMC scores by STL, MTL-Att, MTL-FR and the complete MTL-LOREA on the iLIDS-VID dataset (top) and PRID dataset (bottom).

ture shared by all tasks and task-specific sparse components for individual tasks by MTL. Both low level features and semantic/data-driven attributes are used. We have further proposed a low rank attribute embedding that learns attributes correlations to convert original binary attributes to continuous attributes, where incorrect and incomplete attributes are rectified and recovered. Our objective function can

Table 4.6: CMC scores of ranks from 1 to 50 on the iLIDS-VID and PRID datasets by STL, MTL-Att, MTL-FR and the complete MTL-LOREA. Numbers indicate the percentage (%) of correct matches within a specific rank.

	iLIDS-VID					
Rank	1	5	10	20	30	50
STL	14.7	42.7	41.8	58.5	83.5	91.7
MTL-FR	37.7	54.0	47.4	64.9	85.3	92.5
MTL-Att	40.5	54.9	47.5	64.2	84.2	91.2
MTL-LOREA	**43.0**	**60.0**	**70.2**	**85.3**	**90.2**	**96.3**
	PRID					
Rank	1	5	10	20	30	50
STL	11.3	27.9	41.8	53.0	68.5	74.6
MTL-FR	11.3	34.1	47.4	61.1	69.8	79.0
MTL-Att	12.2	34.7	47.5	61.7	70.9	79.8
MTL-LOREA	**18.0**	**37.4**	**50.1**	**66.6**	**73.1**	**82.3**

be effectively solved by an alternating optimization under proper relaxation. Experiments on four datasets have demonstrated the outstanding performance and robustness of the proposed approach.

Chapter 5: Efficient Object Detection by Deep Neural Networks

5.1 Background

In previous chapters, we have discussed problems related to generic image retrieval and person re-identification, and proposed three approaches to improve the retrieval performance. Besides searching for images containing the same object from the database, it is also critical to detect and recognize objects for better image understanding. In this chapter, we focus on object detection, where the goal is to find and locate instances of specific types of objects, such as cars, pedestrians and animals [1].

Traditionally, designing an object detector involves feature design and choosing learning algorithms, where the two components are usually independent. Any machine learning algorithms can be applied regardless of the type of features used. Designing robust and discriminative hand-crafted features has been an extremely challenging task. Although numerous research works have proposed various kinds of features, the deformable part model (DPM) [17] with hand-crafted features, such as histogram of gradients (HoG), has been the state-of-the-art object detector for decades.

[1] This work was done when the author was an intern in NEC Laboratories America, Inc.

Recently, deep convolutional neural network (CNN) [138, 139] has emerged as a powerful tool that enables end-to-end training/testing and replaces both features design and learning algorithm selection. CNN has contributed much to various computer vision problems including image classification, object detection, semantic segmentation, video recognition, *etc.*, thanks to its capability to learn discriminative features (or representations) at different levels of granularities. A number of recent studies [140, 141] suggest that high level visual semantics (such as motif, parts, or objects) are appearing in the middle of deep architecture which in turn provide strong cues to recognize complex visual concepts. Leveraging on the representational power of CNN, a number of methods are proposed to detect objects in natural images using CNN [7, 142–145]. Although CNN provides highly discriminative features, yet the computational cost still remains too large to detect objects in real time.

In this chapter, we aim to reduce the computational complexity of the CNN model based on the recent Fast RCNN framework [7], as well improving detection accuracy. The scenario here is autonomous driving, which means we only focus on detecting cars, trucks, pedestrians and cyclists, *etc.* Our framework discovers and locates objects in images from a large number of object proposals as input, where the proposals are rectangular bounding boxes of different sizes and aspect ratios. We investigate two new strategies to detect objects accurately and efficiently using deep convolutional neural network: 1) scale-dependent pooling and 2) layer-wise cascaded rejection classifiers. The scale-dependent pooling (SDP) improves detection accuracy by exploiting appropriate convolutional features depending on the scale of the candidate object proposal. The cascaded rejection classifiers (CRC)

effectively utilize convolutional features and eliminate negative object proposals in a cascaded manner, which greatly speeds up the detection while maintaining high accuracy.

5.2 Related Work

5.2.1 CNN for Object Detection

With the exceptional power on image classification, CNN has been applied to object detection and achieves promising results [7, 142, 144, 146–148]. In [146], detection was treated as a regression problem to object bounding box masks. A deep neural network is learned to generate object boxes and then precisely localize them. Erhan *et al.* [144] designed a deep network to propose class-agnostic bounding boxes for generic object detection. Sermanet *et al.* [149] used a regression network pre-trained for classification tasks to predict object bounding boxes in an exhaustive way, which could be computationally expensive. Each bounding box is associated with a confidence score indicating the presence of an object class. Recently, Girshick *et al.* [142] proposed the R-CNN framework that uses a number of object proposals generated by selective search to fine-tune a pre-trained network for detection tasks. Zhang *et al.* [145] extended R-CNN by gradually generating bounding boxes within a search region and imposing a structured loss to penalize localization inaccuracy in network fine-tuning. To reduce the cost of doing forward pass for each proposal in R-CNN, Fast RCNN [7] has been proposed by sharing convolutional features and pooling object proposals only from the last convolutional layer. More recently,

Faster RCNN [150] replaces the object proposals generated by selective search by a region proposal network (RPN) and achieves further speed-up.

5.2.2 Neural Network Cascades

The Viola-Jones cascaded face detector [151] and its extensions [152,153] have been widely used for decades. The idea of eliminating candidates by combining a series of simple features has recently been applied to CNNs. Sun *et al.* [154] presented an ensemble of networks by combining networks that focus on different facial parts for facial point detection. Facial points are first coarsely predicted and then gradually refined by a 3-level cascade of CNNs. Li *et al.* [155] used a shallow detection network with small scale input images to first reject easy non-face samples, and then apply two deeper networks to eliminate more negatives while maintaining a high recall. To further improve detection accuracy, a calibration network is appended after each detection network for bounding box calibration. More recently, Angelova *et al.* [156] combined a tiny deep network and a modified AlexNet to achieve real-time pedestrian detection. The tiny deep network aims to remove a large number of candidates and leave a manageable size of candidates for the large network to evaluate. Our approach is significantly different from prior methods in that we consider cascaded classifiers by utilizing features from different convolutional layers within a single network, that does not introduce any additional computation.

5.2.3 Using Convolutional Features

Rather than using only the outputs from fully-connected (*fc*) layers, a few works exploit features from different convolutional layers, either by concatenating them or by other popular encoding techniques. One of the most representative works is [157], where neuron activations at a pixel of different feature maps are concatenated as a vector as a pixel descriptor (called "Hypercolumn") for precise localization and segmentation. Xu *et al.* [158] extracted convolutional features in the same way and encode these feature vectors by VLAD and Fisher vector for efficient video event detection. In [159], an approach called DeepProposal is presented to generate object proposals in a coarse-to-fine manner. Proposals are first generated in higher level convolutional layers that preserve more semantic information, and are gradually refined in lower layers that provides better localization. Similarly, Karianakis *et al.* [160] used features from lower-level convolutional layers to generate object proposals by sliding window and remove background proposals, while refining them using higher-level convolutional features in a hierarchical way. For edge detection, Bertasius *et al.* [161] extracted a sub-volume from every convolutional layers, perform three types of pooling and again concatenate these values into a single vector, which is further fed into *fc* layers. In contrast to these works, our approach does not explicitly combine convolutional features, but learns classifiers separately.

5.3 Proposed Approach

5.3.1 R-CNN and Fast RCNN

Since our framework is inspired by two recent CNN-based object detectors: R-CNN [142] and Fast RCNN [7], we will first briefly introduce the two detectors, along with their advantages and drawbacks. R-CNN [142] has been proposed for object detection and achieved promising results, where a pre-trained network is fine-tuned to classify thousands of object proposals. However, both training and testing suffer from low efficiency since the network performs a forward pass on every single object proposal independently. Convolutional filters are repeatedly applied to a large number of object proposals, which is computational expensive. In order to reduce the computational cost, recent CNN based object detectors, such as Fast RCNN [7] and Spatial pyramid pooling networks (SPPnet) [143], share the features generated by convolutional layers and apply a multi-class classifier for each candidate proposal. In Fast RCNN, convolutional operations are done only once on the whole image. Features for object proposals are pooled from the feature maps of the last convolutional layer and fed into fully-connected layers (fc) to evaluate the likelihood of object categories. Compared to previous CNN based detector [142], these methods improve efficiency in the order of magnitude via shared convolutional layers. For instance, Fast RCNN achieves $3\times$ and $10 \sim 100\times$ speedup at training and test stage, respectively. In order to deal with scale variation, multi-scale image inputs are often used where one set of convolutional features are obtained per image scale.

Despite its success, these approaches have certain drawbacks that make them less flexible. First, Fast RCNN does not handle small objects well. Since the candidate bounding boxes are pooled directly from the last convolutional feature maps rather than being warped into a canonical size, they may not contain enough information for decision if the boxes are too small. Multi-scale input scheme fundamentally limits the applicability of very deep architecture like [162] due to memory constraints and introduces additional computational burden. In addition, pooling a huge number of candidate bounding boxes and feeding them into high-dimensional *fc* layers can be extremely time-consuming.

5.3.2 Overview of Our Framework

In this work, we address the aforementioned drawbacks and propose a new CNN architecture for accurate and efficient object detection in images. The first contribution is that, unlike previous works, our method produces only one set of convolutional features for an image while handling the scale variation via multiple scale-dependent classifiers. Our intuition is that visual semantic concepts of an object can emerge in different convolutional layers depending on the size of the target objects, if proper supervision is provided in the training process. For instance, if a target object is small, we may observe a strong activation of convolutional neurons in earlier layers (e.g. *conv3*) that encodes specific parts of an object. On the other hand, if a target object is large, the same part concept will emerge in much later layers (e.g. *conv5*). Based on this intuition, we represent a candidate object proposal

(bounding box) using the convolutional features pooled from a layer corresponding to its scale (scale-dependent pooling (SDP)). The pooled features are fed into multiple scale-dependent object classifiers to evaluate the likelihood of object categories. As for the second contribution, we present a novel cascaded rejection classifiers (CRC) where the cascading direction is defined over the convolutional layers in the CNN. We treat the convolutional features in each layer as weak classifiers in the spirit of boosting classifiers [163]. Although the features from the earlier convolutional layers might be too weak to make a strong evaluation of an object category, they are still useful to quickly reject easy negatives. Combining the two strategies, we can explicitly utilize the convolutional features at all layers instead of using only the last one as previous works do. Our method is illustrated in Figure 5.1. We will elaborate the two contributions in the following sections.

5.3.3 Scale-Dependent Pooling

5.3.3.1 Motivation

To handle scale variation, previous works [164, 165] often adopt a sliding window technique with image pyramids to handle scale variation of target objects. Similar techniques are applied in recent CNN based object recognition methods: they treat the last convolutional layer's outputs (*conv5* of AlexNet) as the features to describe an object and apply a classifier (*fc* layers) on top of the extracted features.

R-CNN [142] warps the image patch within a bounding box that produces fixed dimensional feature output for the classification. The independent warping

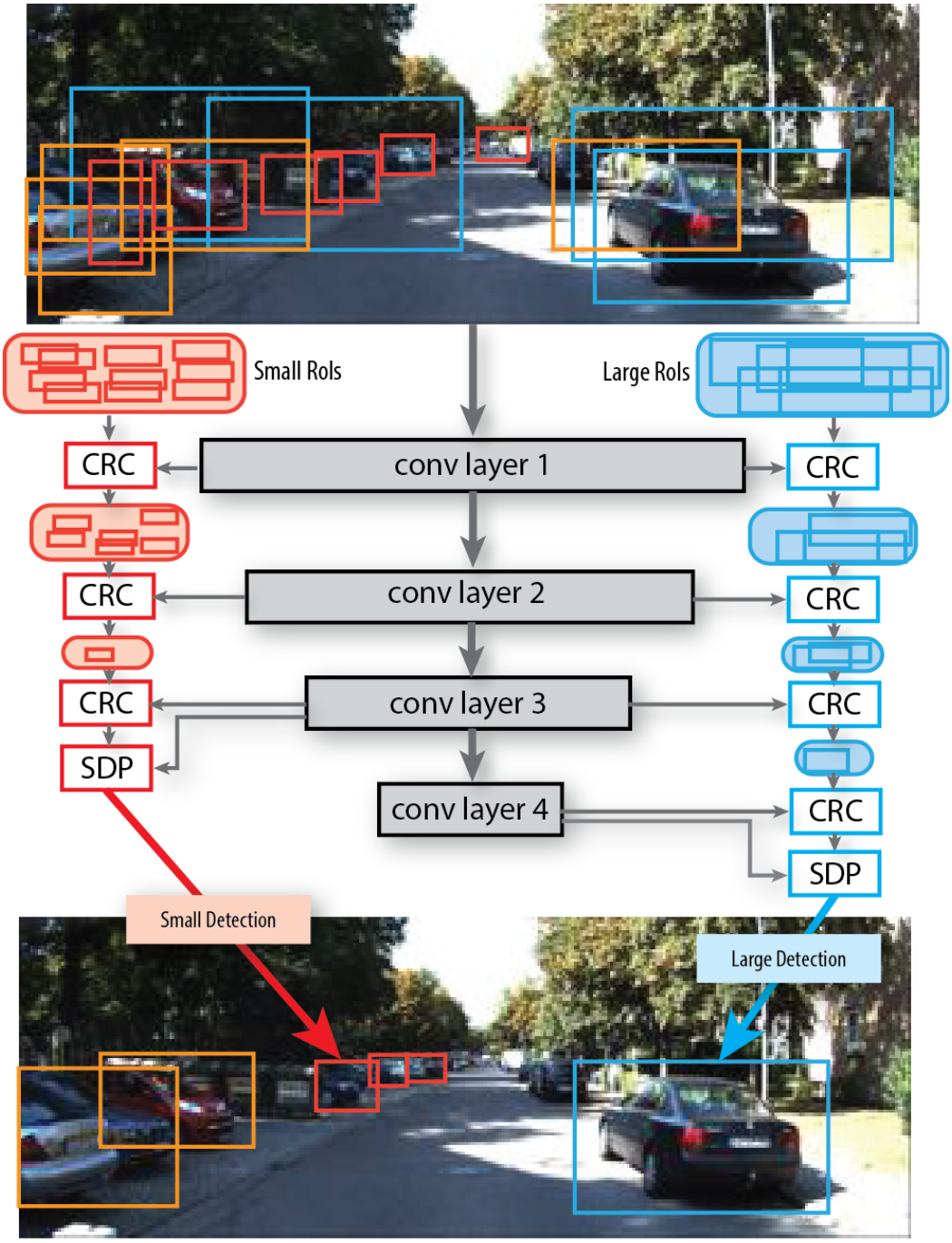


Figure 5.1: We present a fast and accurate object detection method using the convolutional neural network. Our method exploits the convolutional features in all layers to reject easy negatives via *cascaded rejection classifiers* and evaluate surviving proposals using our *scale-dependent pooling* method.

process prohibits us to share any convolutional operations across proposals in the same image, which fundamentally limits the efficiency. In contrast, SPPnet and Fast RCNN [7, 143] share the convolutional features in an image and pool the features at the last convolutional layer to describe an object. In these methods, the scale variation is tackled either via image pyramid inputs or brute-force learning method which directly learns the scale variation via convolutional filters. However, the image pyramid introduces additional computational burden and requires large amount of GPU memories, and brute-force learning via convolutional filters is difficult.

5.3.3.2 Structure of Scale-Dependent Pooling

To alleviate the aforementioned drawbacks of R-CNN and Fast RCNN, we introduce a *scale-dependent pooling* (SDP) technique (illustrated in the Figure 5.2) to effectively handle the scale variation in object detection problem. Our method is built upon Fast RCNN that pools the features for each object proposal from the last convolutional layer of CNN. The region inside of each proposal is divided into a spatial grid (7×7 or 6×6) and features are pooled using max-pooling over each grid. Our SDP method examines the scale (height) of each object proposal and pools the features from a corresponding convolutional layer depending on the height. For instance, if an object proposal has a height between 0 to 64 pixels, the features are pooled from the 3^{rd} convolutional layer of CNN (SDP_3). On the other hand, if an object proposal has a height larger than 128 pixels, we pool the features from the last convolutional layer (SDP_5) (see Figure 5.2). The fully-connected layers attached

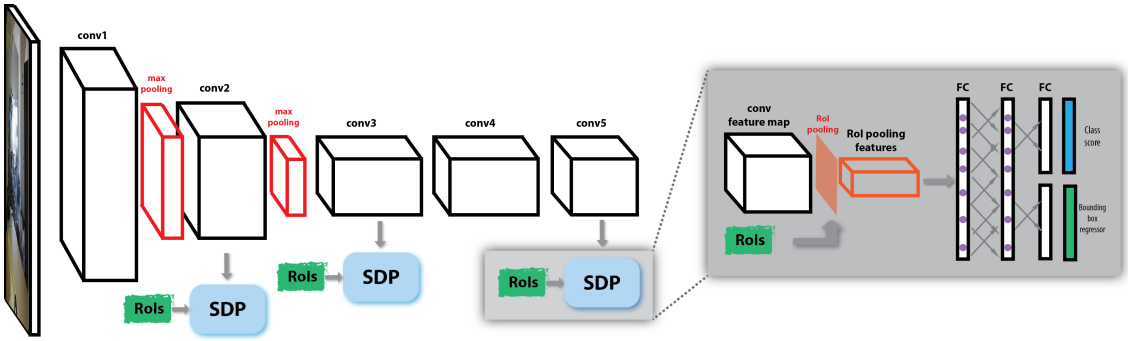


Figure 5.2: Details of our scale-dependent pooling (SDP) model on 16-layer VGG net (VGG16). For better illustration, we show the groups of convolutional filters between max pooling layers as a cube, where filters are arranged side-by-side, separated by lines.

to SDPs have their own set of parameters so as to learn scale-specific classification models from different sets of feature inputs.

We present our SDP model based on VGG16 [162] in Figure 5.2. This SDP model has 3 branches after *conv2*, *conv3* and *conv5*, denoted as SDP_2, SDP_3 and SDP_5. Each branch consists of a *RoI pooling* layer connected to 2 successive *fc* layers with *ReLU* activations and *dropout* layers for calculating class scores and bounding box regressors, similarly to [7]. We initialize the model parameters of convolutional layers and the *fc* layers in the SDP_5 with the ImageNet pre-trained model of VGG16 [162], while the *fc* layers in the SDP_2 and SDP_3 are randomly initialized. During the fine-tuning, input object proposals are first distributed into 3 groups based on their height and then fed into corresponding *RoI pooling* layer to pool the features from corresponding convolutional outputs. Gradients are back-propagated from 3 branches to update corresponding *fc* layers and convolutional filters. By providing supervision about the scale of input object proposals, we explicitly enforce neurons to learn for different scales of objects, so that the convolutional

layers are able to discover small objects at an early stage.

5.3.3.3 Advantages of Scale-Dependent Pooling

The main benefit of SDP is that we can effectively tackle the scale variation in target objects while computing the convolutional features only once per image. Instead of artificially resizing the image inputs in order to obtain a proper feature description as in the image pyramid technique, the SDP selects a proper feature layer to describe an object proposal. It helps us to save additional computational cost and memory overhead caused by redundant convolutional operations.

Another benefit is that the SDP enables us to have a compact and consistent representation of object proposals. Since the brute-force approach of Fast-RCNN [7] pools the features for object proposals from the last convolutional layer, often the same convolutional features are repeated over the spatial grid if an object proposal is very small. The max-pooling or multiple pixel stride in convolutional layers progressively reduces the spatial resolution of the convolutional features over layers. Thus, at the last convolutional layer, there are only one feature corresponding to a large number of pixels (16 pixels for both AlexNet [138] and VGG16 [162]). In the extreme case, if the object proposal is as small as 16×16 pixels, all the grid features may be filled with a repeating single convolutaional feature value. Learning from such an irregular description of object examples may prohibit us from learning a strong classification model. Since the SDPs distribute the proposals depending on the scale, we can provide more consistent signal through the learning process, which

leads to a better detection model.

The idea of using intermediate convolutional layers to complement high level convolutional features has also been recently exploited for image classification and segmentation [157, 166], video event detection [158] and image retrieval [167, 168]. We note that our approach is different from previous works in that we are not simply combining convolutional features from different layers, but adding additional *fc* layers on top of convolutional layers to enforce the neurons to learn scale-specific patterns during the training process.

5.3.4 Cascaded Rejection Classifiers

5.3.4.1 Motivation

One major computational bottleneck in our SDP method and Fast RCNN [7] framework is on the evaluation of individual bounding box proposals using high dimensional *fc* layers. When there are thousands or tens of thousands of object proposals, time spent for the per-proposal evaluation dominates in the entire detection process (see Table 5.5). Therefore, we introduce a novel cascaded rejection classifier (CRC) scheme that requires minimal amount of additional computation. Cascaded detection framework has been widely adopted in visual detection problems that include [151–153]. The core idea is to use as little computation as possible to reduce object proposals quickly and use complex and time-consuming features for only few highly likely candidate proposals. Recently, a few methods [154–156] are proposed to use cascaded detection framework with CNN, but most of them em-

ploy another shallower network to "preprocess" object proposals and use a deeper architecture to evaluate surviving candidates. Unlike the others, we exploit the convolutional features in earlier layers to build the cascaded rejection classifiers. Our model does not require any additional shallow networks or additional convolutional feature computation.

5.3.4.2 Learning Cascaded Rejection Classifiers

We adopt the popular discrete AdaBoost [163] algorithm to learn CRCs after each convolutional layer. Following the intuition of our SDP models, we learn separate rejection classifiers per scale-group ($\mathcal{R}_s^l$ where s and l represent a scale-group and the convolutional layer) in order to keep the classifiers compact while effective (see Figure 5.3). In the following paragraphs, we assume that we have a CNN model trained with SDPs without loss of generality. The rejection threshold of each $\mathcal{R}_s^l$ is trained to keep 99% of positive examples using 50 weak-learners.

Let us first define necessary notations to learn a CRC $\mathcal{R}_s^l$. Suppose we have N proposals belonging to a scale group s, $\mathcal{B} = [B_1, B_2, ..., B_N]$ and corresponding foreground label $y_i, i = 1, ..., N$. $y_i = 1$ if it contains a foreground object and $y_i = 0$, otherwise. We pool the corresponding features x_i for $B_i \in \mathcal{B}$ from convolutional layer l using the CNN model trained with our SDPs. In our experiments, we use the *RoI Pooling* scheme of [7], which gives $m \times m \times c$ dimensional features, where $m = 7$ and c is the number of channels in the convolutional layer. Through this process, we obtain a training dataset of $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{m^2 c \times N}$, and $\mathbf{Y} = \{0, 1\} \in \mathbb{R}^N$.

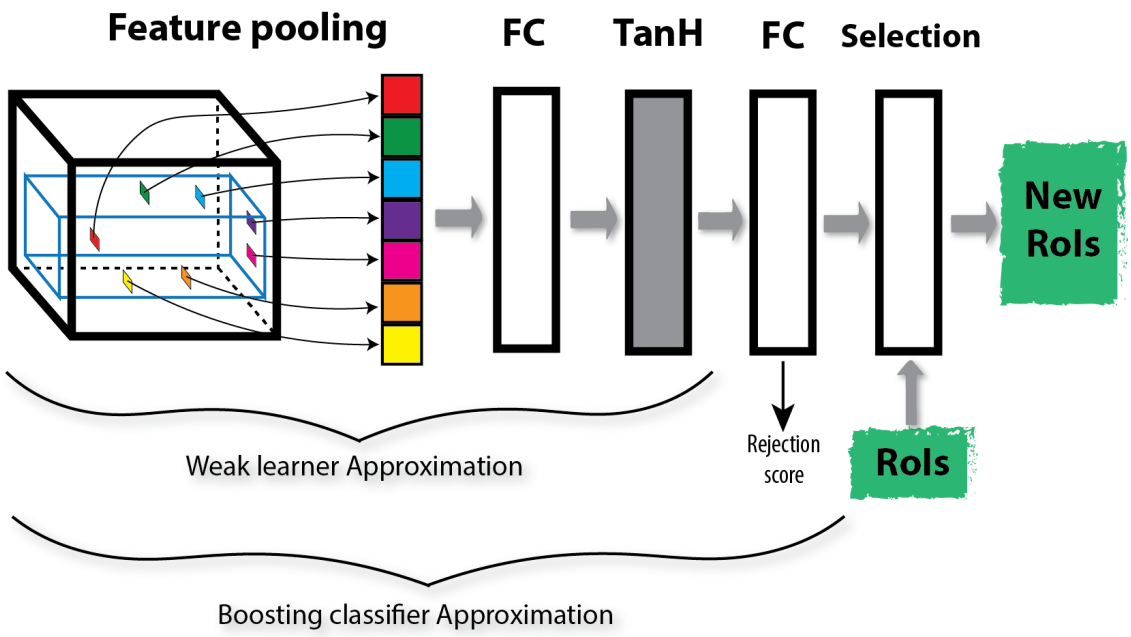


Figure 5.3: Structure of the rejection classifier approximation by network layers. Blue cuboid corresponds to a proposal on the feature maps. Color squares are feature points that need to be pooled out to form the feature vector.

Given the training dataset, we learn a linear boosting classifier $\mathcal{R}_s^l$ with [163] that aggregates a set of weak learners' responses, $\mathcal{R}_s^l(\mathbf{x}) = \sum_{t=1}^{T} w_t h_t(\mathbf{x})$, where h_t is a weak learner, w_t is the corresponding weight and the output is the classification score. In this work, a weak learner h_t is a decision stump that outputs 1 if the value x_v at the v^{th} feature dimension is greater than a decision threshold δ_v and -1 otherwise, that can be written as $h_t(\mathbf{x}) = sign(x_v - \delta_v)$. We learn 50 weak-learners per $\mathcal{R}_s^l$. After learning the boosting classifier, we train the rejection threshold that keeps 99% of positive training examples. All surviving training examples are passed to train the rejection classifier $\mathcal{R}_s^{l+1}$ in the next layer. In order to learn progressively stronger rejection classifier without additional computational cost, the weak learners used in the previous $\mathcal{R}_s^l$ are used to initialize the boosting classifier $\mathcal{R}_s^{l+1}$ in the next

layer.

5.3.4.3 Cascaded Rejection Classifiers in Testing

Since we know which features must be pooled after training the CRCs, we pool only the necessary features in the testing time. We implement a *feature pooling* layer (see Figure 5.3) that pools the convolutional features at specific locations in the feature maps according to trained boosting classifiers. The pooled features are then rearranged as a feature vector. Given the 50 dimensional pool of weak learners, we approximate the boosting classifier with 2 *fc* layers and a hyperbolic tangent *tanh* layer, so as to utilize the computational modules in the CNN framework. The first *fc* layer applies the translation of the features with δ_v, which is followed by the *tanh* layer that approximates the *sign* function. In this way, we successfully approximate the behavior of weak learners by neural network layers. Finally, all the weak learners are aggregated via the last *fc* layer to produce the final boosting classification score using w. If available ($l > 1$), the previous rejection classifier $\mathcal{R}_s^{l-1}$ score is added to the output of the current classifier $\mathcal{R}_s^l$ before rejecting an object proposal. The detailed structure of the CRC is illustrated in Figure 5.3. We observe that the cascaded rejection classifiers achieve about 3.2× speedup for the proposal evaluation (4.6× when combined with truncated SVD [7], see Table. 5.5) with a marginal loss of accuracy.

5.4 Experiments

5.4.1 Experimental Setup

5.4.1.1 Datasets

We evaluate our model with SDP and CRC on two datasets: KITTI detection benchmark [169] and a newly collected Inner-city dataset. The KITTI dataset is composed of 7481 images for training, and 7518 images for testing. The training dataset contains 28742, 4487, and 1627 number of car, pedestrian and cyclist annotations. Since the groundtruth annotation of testing set is not publicly available, we use the training/validation split of [170] for the analysis. For more thorough analysis, we have collected a new dataset (Inner-city). The dataset contains 24509 images which are collected using a camera mounted on a car. The dataset is composed of 16028 training and 8481 testing images which contains 60658, 36547, 16842, and 14414 numbers of car, person, bike and truck instances, respectively. The images are sub-sampled 15 frames apart from 47 number of video sequences to avoid having highly correlated images.

5.4.1.2 Networks

Our CNN model is initialized with a deep network architecture (VGG16 [162]) trained on the ImageNet classification dataset [171]. Rather than having SDP branches for all convolutional layers, we add 3 SDP branches after 3 convolutional layers before max pooling, which are *conv3_3* (SDP_3), *conv4_3* (SDP_4) and *conv5_3*

(SDP_5) of VGG16, to ensure the features are discriminative enough. We use scale groups of height between $[0, 64)$ for SDP_3, $[64, 128)$ for SDP_4, and $[128, \infty)$ for SDP_5. The *fc* layers in the SDP_5 are initialized with the pre-trained model parameters, while the *fc* layers in the SDP_3 and SDP_4 are randomly initialized. All the *fc* layers have 4096 dimensional outputs. After fine-tuning, we train rejection classifiers for each scale group using the convolutional features from *conv1_2*, *conv2_2*, *conv3_3*, *conv4_3* and *conv5_3*, resulting in 12 rejection classifiers.

5.4.1.3 Training Parameters

Following the procedure introduced in [7], we randomly sample two images, from which we randomly sample 128 positive and negative object proposals per scale group as a minibatch. The negative object proposals are sampled from all the proposals that have less than 0.5 overlap with any positive groundtruth annotation. For all the experiments, we use initial learning rate of 0.0005 and decrease it by 0.1 after every 30K iterations. We use the momentum 0.9 and the weight decay 0.0005. The final model is obtained after 90K iterations. We found that using smaller dropout ratio helps to improve the detection accuracy in our experiments, so we use a dropout ratio 0.25 after *fc* layers for all the experiments. For boosting rejection classifiers, we use 50 weak learners corresponding to 50 locations in the feature maps.

As for object proposals, we obtain the bounding box proposals using Edge-box [172] and augment them with ACF [165] detection outputs trained for Car and

Person categories. We observe that using only generic box proposal methods often misses small target objects, which leads to poor detection accuracy.

5.4.2 Detection Results

We first discuss the detection accuracy on the KITTI train/validation dataset and the Inner-city dataset. We mainly compare our model against two baselines using Fast RCNN models [7] with AlexNet [138] and VGG16 [162] architectures. For the KITTI train/validation experiment, all the training and testing images are rescaled to 500 pixel height which produces the best accuracy given GPU (K40/K80) memory constraints. Since AlexNet architecture consumes much less memory, we use multi-scale image inputs of $400, 800, 1200, 1600$ pixel heights input for the AlexNet baseline to handle scale variation as well as possible. In the Inner-city experiments, we keep the original size of images (420 pixel height) for the VGG16 baseline and use $420, 840, 1260, 1680$ pixels for the AlexNet baseline. In order to highlight the challenges posed by scale variation, we present the accuracy comparison over different size groups in Table 5.1 and 5.2. Following KITTI [169] evaluation protocol, we use 0.7 overlap ratio for the Car category and 0.5 for the others in the evaluation. In the Inner-city evaluation, we use 0.5 overlap ratio across all categories.

5.4.2.1 Results by SDP

Table 5.1 and 5.2 show that the multi-scale image input baseline with AlexNet architecture (FRCN [7]+AlexNet) achieves similar detection accuracy across differ-

ent scale groups, since features are pooled at appropriate scales. On the other hand, deeper architecture with a single image input baseline (FRCN [7]+VGG16) achieves higher accuracy on larger objects exploiting the rich semantic features, but performs relatively poorly on small objects. We believe this is due to the difficulty in learning visual concepts at various scales via a single final layer. In contrast, our SDP model with the same VGG16 architecture achieves highest accuracy on almost all scale groups over all the categories.

More importantly, we greatly improve the detection accuracy on the smallest scale group by $5 \sim 20\%$ thanks to the use of SDP branches attached to the intermediate convolultional layers, which confirms our hypothesis that small objects can be better recognized at lower layers if proper supervision is provided in the training process. Another important observation is that we achieve larger improvement on the Car category which has the largest number of training examples. Since our model has additional parameters to be trained (*fc* parameters in SDP_3 and SDP_4), we expect that our model will improve even more when more training examples are provided. This is demonstrated in the experiments on Inner-city dataset (see Table 5.1 and 5.2) that contains larger number of training examples. A few qualitative results are presented in Figure 5.4.

5.4.2.2 Results by CRC

Next, we evaluate the performance of our cascaded rejection classifiers (CRC). As described in Section 5.3.4, we reject object proposals through our CRCs through-

Table 5.1: Detection AP (%) of baselines and our models on KITTI validation set, divided by size groups. S_1, S_2, S_3, S_4 and S indicate the size group of $[0, 64)$, $[64, 128)$, $[128, 256)$, $[256, \infty)$ and $[0, \infty)$. We use 4 scale image pyramid for FRCN [7]+AlexNet and 1 scale image input for the others.

		FRCN [7]+AlexNet	FRCN [7]+VGG16	SDP	SDP+CRC	SDP+CRC *ft*
	Inputs	4	1	1	1	1
Car	S_1	52.8	42.2	**64.2**	63.9	63.9
	S_2	60.7	70.0	**74.4**	74.3	74.2
	S_3	75.8	85.1	**86.0**	85.8	85.5
	S_4	55.5	65.9	**68.4**	68.2	62.9
	S	61.6	62.3	**73.7**	73.5	73.7
Pedestrian	S_1	**19.7**	12.6	17.3	17.5	17.6
	S_2	47.5	55.9	**58.4**	52.0	50.0
	S_3	88.4	94.6	**94.9**	93.7	93.4
	S_4	24.1	44.9	44.8	45.9	**61.0**
	S	61.4	66.8	**66.9**	65.5	65.9
Cyclist	S_1	**42.0**	29.1	37.5	35.1	35.8
	S_2	51.6	63.8	**67.3**	65.7	66.5
	S_3	44.9	68.7	68.6	**69.2**	67.6
	S_4	0.0	0.0	0.0	0.0	0.0
	S	46.5	48.8	**54.6**	52.9	53.1
mAP	S	56.5	59.3	**65.1**	64.0	64.2

Table 5.2: Detection AP (%) of baselines and our models on the Inner-city dataset, divided by size groups. S_1, S_2, S_3, S_4 and S indicate the size group of $[0, 64)$, $[64, 128)$, $[128, 256)$, $[256, \infty)$ and $[0, \infty)$. We use 4 scale image pyramid for FRCN [7]+AlexNet and 1 scale image input for the others.

		FRCN [7]+AlexNet	FRCN [7]+VGG16	SDP	SDP+CRC	SDP+CRC *ft*
	Inputs	4	1	1	1	1
Car	S_1	74.6	63.9	**76.2**	75.7	75.0
	S_2	78.9	80.0	**84.2**	83.8	84.1
	S_3	82.9	86.4	86.9	86.5	**87.2**
	S_4	94.9	93.7	95.2	95.0	**95.6**
	S	82.4	80.5	**85.5**	85.0	84.9
Pedestrian	S_1	43.9	35.2	**51.1**	50.9	**51.1**
	S_2	69.1	71.3	**78.0**	75.9	76.7
	S_3	77.8	**83.3**	83.0	80.2	80.2
	S_4	75.4	77.3	**81.5**	78.3	77.8
	S	63.7	64.3	**73.9**	71.7	72.2
Bike	S_1	26.2	28.2	40.3	38.4	**41.6**
	S_2	42.3	57.5	**65.4**	61.5	64.6
	S_3	45.9	**68.7**	65.2	63.7	64.7
	S_4	2.2	0.5	43.2	41.5	**46.9**
	S	36.3	50.6	57.9	55.1	**58.2**
Truck	S_1	28.7	26.0	44.1	43.9	**45.8**
	S_2	51.5	62.1	67.0	66.8	**69.1**
	S_3	60.0	70.0	**71.5**	71.0	69.9
	S_4	67.0	54.0	75.1	**75.6**	74.2
	S	48.7	53.6	65.6	65.5	**66.4**
mAP	S	55.0	62.2	**70.7**	69.3	70.4

Figure 5.4: Qualitative results on KITTI validation set and Inner-city dataset using FRCN [7]+VGG16 baseline and our SDP model. We obtain the detection threshold for visualization at the precision 0.8. Notice that our method with SDP detect small objects much better than the baseline method. The figure is best shown in color.

out the convolutional layers. With the CRC modules (denoted as SDP+CRC in Table 5.1 and 5.2), the performance decreases very marginally, indicating that the rejection classifiers successfully eliminate negative proposals while maintaining a high recall rate for positives (see Table 5.4 for details), even though we only use 50 feature dimensions at each convolutional layer. The results demonstrate that the intermediate convolutional layers can be exploited in a hierarchical way.

5.4.2.3 Fine-tuning with CRC

We further fine-tune the network with trained CRC modules to see if it can further improve performance. The CRC modules can serve as a hard-negative mining process to learn better classification model in the network, since many easy negatives are rejected before reaching the SDP modules. Instead of randomly sampling 128 proposals in the training process, we sample 128 proposals from survived proposals after using all the CRCs. We run the fine-tuning for additional 50K iterations with initial learning rate 0.0001 with step size 20K iterations. We freeze the learning rate of convolutional layers to avoid CRC parameters being invalid after the fine-tuning. We observe that the additional fine-tuning (SDP+CRC *ft*) helps to improve the accuracy over the SDP+CRC variant marginally. In KITTI testing results (see Table 5.3), we observe larger improvement with the additional fine-tuning. We believe that it will achieve more improvement, if all the model parameters including CRC modules are trained properly.

5.4.2.4 Test Set Evaluation

To compare with existing approaches on KITTI test set, we train our SDP and CRC models on the full training set, and evaluate it on the test set. The results are shown in Table 5.3. We use the same configuration and learning parameters as in the previous experiments. Without using any stereo information, our approach outperforms all compared methods on all levels of difficulties and achieves the best results. In particular, our method using SDP again outperforms the Fast-RCNN baseline by 9% on average, verifying the effectiveness of the SDP module. Notably, our method improves AP by 16.7% over Fast-RCNN baseline on Hard case of Car category, where most samples are of small size or occluded. This is a clear evidence showing the discriminative power of our SDP module.

5.5 Discussion and Analysis

In this section, we conduct additional experiments to further analyze and better understand the proposed approach.

5.5.1 Rejection Ratio

By using CRCs, we aim to improve the efficiency for the proposal evaluation by progressively reducing the number of proposals. In Table 5.4, we analyze the percentage of surviving proposals with respect to the initial number of input proposals after applying CRCs, as well as the corresponding recall rate of positives after each CRC. The table shows that our CRCs successfully reject a large number of input

Table 5.3: Detection AP (%) of the other state-of-the-art approaches and our method on KITTI test set. Following KITTI protocol, results are grouped into three levels of difficulties: Easy (E), Moderate (M) and Hard (H).

	Car			*Pedestrian*			*Cyclist*		
Method	E	M	H	E	M	H	E	M	H
Regionlet [173]	84.75	76.45	59.70	73.14	61.15	55.21	70.41	58.72	51.83
DPM-VOC+VP [174]	74.95	64.71	48.76	59.48	44.86	40.37	42.43	31.08	28.23
3DVP [170]	87.46	75.77	65.38	-	-	-	-	-	-
SubCat [175]	84.14	75.46	59.71	-	-	-	-	-	-
CompACT-Deep [176]	-	-	-	70.69	58.74	52.71	-	-	-
DeepParts [177]	-	-	-	70.49	58.67	52.78	-	-	-
FRCN [7]+VGG16	85.98	72.32	60.16	75.50	62.53	58.14	68.82	54.21	47.98
SDP	88.34	81.69	69.72	76.89	**64.44**	**59.72**	70.13	60.08	52.93
SDP+CRC	88.33	81.17	70.00	76.28	63.12	58.30	71.06	60.24	53.17
SDP+CRC *ft*	**90.33**	**83.53**	**71.13**	**77.74**	64.19	59.27	**74.08**	**61.31**	**53.97**

proposals while keeping a high recall for the true objects. For each scale group, CRCs can remove over 70 $\sim$ 80% input proposals, so that only around 20 $\sim$ 30% proposals go through *fc* layers that are computationally expensive.

5.5.2 Runtime Efficiency

We investigate the efficiency gain introduced by CRCs. Table 5.5 analyzes detailed computational breakdown of various methods. We measure the time spent

Table 5.4: Percentage (%) of surviving proposals after applying CRC, and the corresponding recall rate (%) on KITTI validation set. $\mathcal{R}_{[n_1,n_2)}$ refers to the rejection classifier for the scale group $[n_1, n_2)$.

	$\mathcal{R}_{[0,64)}$		$\mathcal{R}_{[64,128)}$		$\mathcal{R}_{[128,\infty)}$		Overall	
Layer	ratio	recall	ratio	recall	ratio	recall	ratio	recall
conv1_2	66.2	97.6	83.9	98.1	94.8	100	81.6	98.6
conv2_2	44.2	95.5	59.2	96.2	92.9	99.7	65.4	97.1
conv3_3	16.7	92.1	25.1	93.4	72.3	96.5	38.0	94.0
conv4_3	-	-	12.6	90.3	48.6	92.0	30.6	91.2
conv5_3	-	-	-	-	28.8	89.9	28.8	89.9

in each component of the network, such as convolutional operations, *fc* layer computations, pre- and post-processing, *etc.* We compare our CRCs with the truncated SVD approach [7] that aims to reduce the dimensionality of *fc* layers. We follow the strategy in [7] to keep the top 1024 singular values from the 1^{st} *fc* layer and the top 256 singular values from the 2^{nd} *fc* with respect to each SDP branch. In addition, we combine CRC and SVD, *i.e.*, using CRC to eliminate proposals and SVD to compress *fc* layers in SDPs, to achieve further speed-up. We include the baseline methods without SVD as a reference.

The truncated SVD approach alone achieves about 2.3× gain in proposal evaluations. The CRC modules alone obtain 3.2× speed-up for the same operation. We gain $4 \sim 5\times$ speed-up for each SDP by rejecting $70 \sim 80\%$ of proposals, but the additional computation introduced by CRC reduces the overall gain slightly. When

Table 5.5: Runtime comparison (ms per image) among the baseline methods, our method with truncated SVD [7], our method with CRC and SVD+CRC on KITTI dataset. fc_S, fc_M, and fc_L refer to the SDP classifiers for the scale group [0, 64), [64, 128), [128, ∞), respectively. "box eval." represents the time spent for individual box evaluation including *fc* layers and CRC rejections. The times were measured using an Nvidia K40 GPU under the same experimental environment.

Component	*conv*	*fc*	fc_S	fc_M	fc_L	rej.	box eval.	misc.	total
[7]+AlexNet	799	512	0	0	0	0	512	164	1476
[7]+VGG16	282	719	0	0	0	0	719	21	1022
SDP	286	0	204	254	283	0	741	90	1117
SVD	285	0	97	116	114	0	327	95	707
speedup	1.0	-	2.10	2.19	2.48	-	2.27	0.95	1.58
CRC	282	0	44	46	63	79	232	27	541
speedup	1.0	-	4.64	5.52	4.49	-	3.19	3.33	2.06
SVD+CRC	283	0	24	25	31	81	161	27	471
speedup	1.0	-	8.50	10.16	9.13	-	4.60	3.33	2.37

combining SVD and CRC, we obtain 4.6× efficiency gain in proposal evaluations and 2.4× in total (including *conv* operations).

5.5.3 Speed versus Accuracy

Next, we show the change of detection accuracy and speed with respect to varying rejection ratios. To do this, we use a fixed rejection ratio for each rejection classifier and deactivate the corresponding learned rejection threshold. In particular,

given a number of proposals, we first apply the rejection classifier at a layer to obtain classification scores. Then we rank the proposals based on the classification scores, where proposals with larger scores are ranked higher. All proposals ranked at the bottom $K\%$ are removed regardless of the rejection threshold. The remaining proposals go through to next layers. In the experiments, we set $K = 30, 50, 70, 90$, which means we reject 30%, 50%, 70% or 90% proposals at each layer. We evaluate both our CRCs and CRCs+SVD variants, and compare them with baselines in terms of accuracy and running speed in Figure 5.5.

We observe that even we reject 30% proposals at each layer, which results in removing 83% proposals totally for the largest size group of proposals, we still achieve reasonable accuracy. While using more aggressive rejection ratios speeds up the detection, it greatly affects the accuracy. By learning proper rejection thresholds from trained CRCs, we obtain a good trade-off between detection accuracy and detection speed without explicitly tuning the rejection ratio.

5.6 Summary

In this chapter, we have investigated two new strategies to detect objects efficiently using deep convolutional neural network, 1) scale-dependent pooling and 2) layer-wise cascaded rejection classifiers. The scale-dependent pooling (SDP) improves detection accuracy especially on small objects by fine-tuning a network with scale-specific branches attached after several convolutional layers. The cascaded rejection classifiers (CRC) effectively utilize convolutional features and eliminate neg-

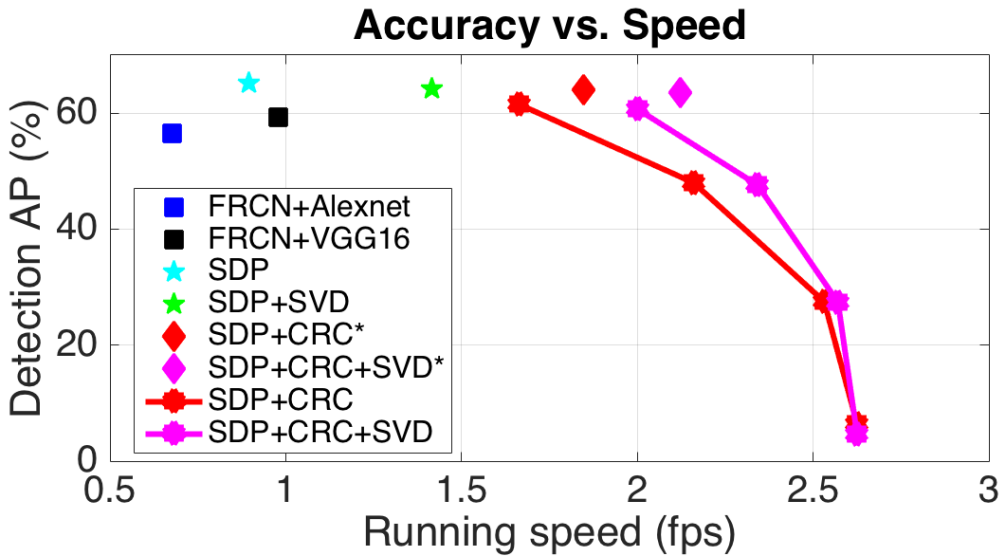


Figure 5.5: Detection AP (%) vs. running speed (fps) with respect to different variants of our SDP models and other baselines on KITTI validation set. SDP and SDP+SVD indicate our SDP model with VGG16, and the same model after applying truncated SVD. SDP+CRC* and SDP+CRC+SVD* indicate the SDP models using CRCs with pre-trained rejection thresholds at each layer. SDP+CRC and SDP+CRC+SVD denote the SDP models using CRCs with varying rejection ratio fixed at each layer.

ative object proposals in a cascaded manner, which greatly speeds up the detection while maintaining high accuracy. Our experimental evaluation clearly demonstrates the benefits of SDP and CRC in CNN based object detection.

Chapter 6: Conclusion

Image retrieval and matching is an important topic in computer vision and has various practical applications, which involves searching and locating for same/similar objects in images. With multiple features available, how to effectively combine them to achieve better results remains a challenging problem. In this work, we focused on leveraging multiple features to improve performance and reduce computational cost with respect to two applications: content-based image retrieval and reranking, and object detection in images. We proposed several approaches to achieve this goal.

(1) We have proposed a supervised multi-feature fusion algorithm based on graphical models for generic image retrieval. We employ a mixture Markov model based on a random walk model on multiple graphs to fuse graphs. We also introduce a probabilistic model to compute the importance of each feature for graph fusion under a naive Bayesian formulation, and employ an iterative diffusion algorithm alleviate the effect of noise.

(2) To reduce human labeling, we have proposed a fully unsupervised reranking approach based on a submodular objective function that consists of two terms: an information gain term and a relative ranking consistency term. We select a subset from initially retrieved images by maximizing the mutual information (information

gain) between the selected subset and unselected nodes in graph representations. The relative ranking consistency term exploits the inter-relationships among multiple ranked lists obtained by different features. The final submodular objective function combines both the relationships among retrieved images from a single feature and the relative ranks of images across different features, thereby improving initial retrieval results obtained by multiple independent features.

(3) We have then studied a practical application of generic image retrieval: person re-identification, where the database usually contains well labeled data that allows more sophisticated learning algorithms. We have applied the multi-task learning algorithm using both low level features and attributes. A low rank attribute embedding has been introduced into the multi-task learning formulation to embed original binary attributes to a continuous attribute space, where incorrect and incomplete attributes are rectified and recovered to better describe people. Re-identifications from multiple cameras are regarded as related tasks to exploit shared information to improve re-identification accuracy. Specifically, we propose a novel multi-task learning with low rank attribute embedding framework for person re-identification.

(4) To accurately locate objects in images, We have proposed an object detector based on deep convolutional neural networks (CNN). We improve the recent Fast RCNN framework and investigate two new strategies to detect objects accurately and efficiently: 1) scale-dependent pooling and 2) layer-wise cascaded rejection classifiers. The scale-dependent pooling (SDP) improves detection accuracy by exploiting appropriate convolutional features depending on the scale of input

object proposals. The cascaded rejection classifiers (CRC) effectively utilize convolutional features and eliminate negative proposals in a cascaded manner, which greatly speeds up the detection while maintaining high accuracy. In combination of the two, our method achieves significantly better accuracy compared to other state-of-the-arts in two challenging datasets, while being more efficient.

Appendix A: Proof of Propositions

A.1 Proof of PROPOSITION 1

A.1.1 Monotonicity

Proof. We have

$$F_m(\mathcal{S}) = H(\mathcal{V}_m \backslash \mathcal{S}) - H(\mathcal{V}_m \backslash \mathcal{S} | \mathcal{S}) = I(\mathcal{V}_m \backslash \mathcal{S}; \mathcal{S})$$

for graph $\mathcal{G}_m$, where $I(\mathcal{V}_m \backslash \mathcal{S}; \mathcal{S})$ is the mutual information between $\mathcal{V}_m \backslash \mathcal{S}$ and $\mathcal{S}$. As proved in [51], $I(\mathcal{V}_m \backslash \mathcal{S}; \mathcal{S})$ is monotonic when $|\mathcal{V}_m|$ is larger than $2|\mathcal{S}|$, which is the case in our framework. This completes the proof of the monotonicity property of $F_m(\mathcal{S})$. □

A.1.2 Submodularity

Proof. We prove the submodularity by showing: for any $\mathcal{S}_1 \subset \mathcal{S}_2$ and a given example $a \in \mathcal{V}_m \backslash \mathcal{S}_2$, we have

$$F_m(\mathcal{S}_1 \cup \{a\}) - F_m(\mathcal{S}_1) \geq F_m(\mathcal{S}_2 \cup \{a\}) - F_m(\mathcal{S}_2)$$

We have

$$
\begin{aligned}
&(F_m(\mathcal{S}_1 \cup \{a\}) - F_m(\mathcal{S}_1)) - (F_m(\mathcal{S}_2 \cup \{a\}) - F_m(\mathcal{S}_2)) \\
&= (H(a|\mathcal{S}_1) - H(a|\mathcal{V}_m \backslash \{\mathcal{S}_1 \cup a\})) \\
&\quad - (H(a|\mathcal{S}_2) - H(a|\mathcal{V}_m \backslash \{\mathcal{S}_2 \cup a\})) \\
&= (H(a|\mathcal{S}_1) - H(a|\mathcal{S}_2)) \\
&\quad + (H(a|\mathcal{V}_m \backslash \{\mathcal{S}_2 \cup a\}) - H(a|\mathcal{V}_m \backslash \{\mathcal{S}_1 \cup a\})) \\
&= H_1 + H_2
\end{aligned}
$$

Since conditioning always reduces entropy, $H(a|\mathcal{S}_1) \geq H(a|\mathcal{S}_2)$, so that $H_1 \geq 0$. $\mathcal{V}_m \backslash \{\mathcal{S}_2 \cup a\} \subset \mathcal{V}_m \backslash \{\mathcal{S}_1 \cup a\}$, so that we have $H(a|\mathcal{V}_m \backslash \{\mathcal{S}_2 \cup a\}) \geq H(a|\mathcal{V}_m \backslash \{\mathcal{S}_1 \cup a\})$, leading to $H_2 \geq 0$. Therefore, $H_1 + H_2 \geq 0$, which completes the proof of the submodularity property of $F_m(\mathcal{S})$. □

A.2 Proof of PROPOSITION 2

A.2.1 Monotonicity

Proof. We prove that $T(\mathcal{S})$ is monotonically increasing by showing $T(\mathcal{S} \cup \{a\}) \geq T(\mathcal{S})$, for all $a \in \mathcal{V} \backslash \mathcal{S}$ and $\mathcal{S} \subseteq \mathcal{V}$. Let $|\mathcal{S}|$ denote the cardinality of $\mathcal{S}$. Since items in $\mathcal{S}$ are ordered, we assume the rank of a in $\mathcal{S} \cup \{a\}$ as $r_a = |\mathcal{S}| + 1$ without loss of

generality. We have

$$
\begin{aligned}
&T(\mathcal{S}\cup\{a\})-T(\mathcal{S})\\
=&(1-q)\sum\nolimits_{s=1}^{|\mathcal{S}|+1}q^s\cdot\frac{1}{s}\sum\nolimits_{v_i,v_j\in\mathcal{S}\cup\{a\},r_{v_i}<r_{v_j}=s}\mathcal{C}(v_i,v_j)\\
&-(1-q)\sum\nolimits_{s=1}^{|\mathcal{S}|}q^s\cdot\frac{1}{s}\sum\nolimits_{v_i,v_j\in\mathcal{S},r_{v_i}<r_{v_j}=s}\mathcal{C}(v_i,v_j)\\
=&(1-q)\cdot q^{|\mathcal{S}|+1}\cdot\frac{1}{|\mathcal{S}|+1}\sum\nolimits_{v_i\in\mathcal{S},r_{v_i}<r_a=|\mathcal{S}|+1}\mathcal{C}(v_i,a)
\end{aligned}
$$

Since $\mathcal{C}(v_i,a)\geq 0$, $1-q>0$ and $q^{|\mathcal{S}|+1}>0$, we can easily have $T(\mathcal{S}\cup\{a\})-T(\mathcal{S})\geq 0$ and $T(\emptyset)=0$. This completes the proof of monotonically increasing property of $T(\mathcal{S})$. □

A.2.2 Submodularity

Proof. We prove the submodularity by showing: for any $\mathcal{S}_1\subset\mathcal{S}_2$ and a given example $a\in\mathcal{V}\backslash\mathcal{S}_2$, we have

$$
T(\mathcal{S}_1\cup\{a\})-T(\mathcal{S}_1)\geq T(\mathcal{S}_2\cup\{a\})-T(\mathcal{S}_2)
$$

From the derivation for monotonicity, we have

$$
\begin{aligned}
&T(\mathcal{S}_1\cup\{a\})-T(\mathcal{S}_1)\\
=&(1-q)\cdot q^{|\mathcal{S}_1|+1}\cdot\frac{1}{|\mathcal{S}_1|+1}\sum\nolimits_{v_i\in\mathcal{S}_1,r_{v_i}<r_a=|\mathcal{S}_1|+1}\mathcal{C}(v_i,a)
\end{aligned}
$$

and

$$\begin{aligned}&T(\mathcal{S}_2 \cup \{a\}) - T(\mathcal{S}_2)\\ =&(1-q)\cdot q^{|\mathcal{S}_2|+1}\cdot\frac{1}{|\mathcal{S}_2|+1}\sum\nolimits_{v_i\in\mathcal{S}_2, r_{v_i}<r_a=|\mathcal{S}_2|+1}\mathcal{C}(v_i,a)\end{aligned}$$

For notational simplicity, we let $n_1 = |\mathcal{S}_1| + 1$ and $n_2 = |\mathcal{S}_2| + 1$. Define

$$\begin{aligned}k_1 &= \frac{1}{n_1}\sum\nolimits_{v_i\in\mathcal{S}_1, r_{v_i}<r_a=n_1}\mathcal{C}(v_i,a)\\ k_2 &= \frac{1}{n_2}\sum\nolimits_{v_i\in\mathcal{S}_2, r_{v_i}<r_a=n_2}\mathcal{C}(v_i,a)\end{aligned}$$

as the average relative ranking measure between a and all items in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Then k_1 and k_2 can be represented as

$$k_2 = \frac{1}{n_2}(n_1k_1 + \sum\nolimits_{v_i\in\mathcal{S}_2\setminus\mathcal{S}_1, r_{v_i}<r_a=n_2}\mathcal{C}(v_i,a))$$

Suppose $|\mathcal{S}_2| = |\mathcal{S}_1|+n$, $\mathcal{C}(v_i,a)$ can be considered as a random variable $\phi\in[0,1]$, so that we have $k_2 = \frac{1}{n_2}(n_1k_1 + \sum_n \phi)$, where the upper bound of $\sum_n \phi$ is nk_1. Hence

$$\begin{aligned}&(T(\mathcal{S}_1 \cup \{a\}) - T(\mathcal{S}_1)) - (T(\mathcal{S}_2 \cup \{a\}) - T(\mathcal{S}_2))\\ =&(1-q)\cdot q^{|\mathcal{S}_1|}(k_1 - q^n k_2)\end{aligned}$$

Since $(1-q) > 0$ and $q^{|\mathcal{S}_1|} > 0$, we only need to prove $k_1 - q^n k_2 \geq 0$. Let $k_1 - q^n k_2 = k_1 - q^n\frac{n_1k_1+\sum_n\phi}{n_2}$, which reaches its minimum when $\sum_n \phi$ reaches its upper bound.

In this case, we have

$$k_1 - q^n k_2 = k_1 - q^n \frac{n_1 k_1 + n k_1}{n_2} = k_1(1 - q^n) \geq 0$$

This completes the proof of submodularity property of $T(\mathcal{S})$. □

Bibliography

[1] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1–8, 2007.

[2] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[3] Ondrej Chum, Andrej Mikulík, Michal Perdoch, and Jiri Matas. Total recall II: Query expansion revisited. In *CVPR*, pages 889–896, 2011.

[4] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8, 2007.

[5] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.

[6] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008.

[7] Ross Girshick. Fast R-CNN. *arXiv preprint arXiv:1504.08083*, 2015.

[8] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[9] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[10] Xiaoyu Wang, Ming Yang, Timothée Cour, Shenghuo Zhu, Kai Yu, and Tony X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV*, pages 209–216, 2011.

[11] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. Large-scale image retrieval with compressed Fisher vectors. In *CVPR*, pages 3384–3391, 2010.

[12] Matthijs Douze, Arnau Ramisa, and Cordelia Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, pages 745–752, 2011.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[14] Hervé Jégou and Ondrej Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, pages 774–787, 2012.

[15] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1704–1716, 2012.

[16] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, pages 1578–1585, 2013.

[17] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[18] Fan Yang, Bogdan Matei, and Larry S. Davis. Re-ranking by multi-feature fusion with diffusion for image retrieval. In *WACV*, pages 572–579, 2015.

[19] Fan Yang, Zhuolin Jiang, and Larry S. Davis. Submodular reranking with multiple feature modalities for image retrieval. In *ACCV*, pages 19–34, 2014.

[20] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S. Davis, and Wen Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *ICCV*, pages 3739–3747, 2015.

[21] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *To appear in CVPR*, 2016.

[22] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[23] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391, 2010.

[24] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *CVPR*, pages 1169–1176, 2009.

[25] Stefan Romberg and Rainer Lienhart. Bundle min-hashing. *IJMIR*, 2(4):243–259, 2013.

[26] Jérome Revaud, Matthijs Douze, and Cordelia Schmid. Correlation-based burstiness for logo retrieval. In *ACM Multimedia*, pages 965–968, 2012.

[27] Meng Wang, Hao Li, Dacheng Tao, Ke Lu, and Xindong Wu. Multimodal graph-based reranking for web image search. *IEEE Transactions on Image Processing*, 21(11):4649–4661, 2012.

[28] Shaoting Zhang, Ming Yang, Timothée Cour, Kai Yu, and Dimitris N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, pages 660–673, 2012.

[29] Cheng Deng, Rongrong Ji, Wei Liu, Dacheng Tao, and Xinbo Gao. Visual reranking through weakly supervised multi-graph learning. In *ICCV*, pages 2600–2607, 2013.

[30] Shiliang Zhang, Ming Yang, Xiaoyu Wang, Yuanqing Lin, and Qi Tian. Semantic-aware co-indexing for image retrieval. In *ICCV*, pages 1673–1680, 2013.

[31] Liang Zheng, Shengjin Wang, Ziqiong Liu, and Qi Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *CVPR*, 2014.

[32] Aniruddha Kembhavi, Behjat Siddiquie, Roland Miezianko, Scott McCloskey, and Larry S. Davis. Incremental multiple kernel learning for object recognition. In *ICCV*, pages 638–645, 2009.

[33] Mehmet Gönen and Ethem Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[34] Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.

[35] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis. In *ICCV*, pages 24–31, 2009.

[36] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, pages 1473–1480, 2002.

[37] Yi Yang, Jingkuan Song, Zi Huang, Zhigang Ma, Nicu Sebe, and Alexander G. Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Transactions on Multimedia*, 15(3):572–581, 2013.

[38] Yi Yang, Yueting Zhuang, Dong Xu, Yunhe Pan, Dacheng Tao, and Stephen J. Maybank. Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning. In *ACM Multimedia*, pages 311–320, 2009.

[39] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang. Robust late fusion with rank minimization. In *CVPR*, pages 3021–3028, 2012.

[40] Basura Fernando, Élisa Fromont, Damien Muselet, and Marc Sebban. Discriminative feature fusion for image classification. In *CVPR*, pages 3434–3441, 2012.

[41] Xingwei Yang, Suzan Köknar-Tezel, and Longin Jan Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, pages 357–364, 2009.

[42] Dengyong Zhou and Christopher J. C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, pages 1159–1166, 2007.

[43] David Harel and Yehuda Koren. Clustering spatial data using random walks. In *KDD*, pages 281–286, 2001.

[44] Li Fei-Fei, Robert Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[45] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc J. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, pages 777–784, 2011.

[46] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[47] Matthijs Douze, Herve Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009.

[48] Andrej Mikulík, Michal Perdoch, Ondrej Chum, and Jiri Matas. Learning a fine vocabulary. In *ECCV*, pages 1–14, 2010.

[49] Danfeng Qin, Christian Wengert, and Luc Van Gool. Query adaptive similarity for large scale object retrieval. In *CVPR*, 2013.

[50] Giorgos Tolias, Yannis S. Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, pages 1401–1408, 2013.

[51] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[52] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *CVPR*, pages 1897–1904, 2011.

[53] Gunhee Kim, Eric P. Xing, Fei-Fei Li, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, pages 169–176, 2011.

[54] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *CVPR*, pages 2097–2104, 2011.

[55] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(1):99–112, 2014.

[56] Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *ICML*, pages 567–574, 2010.

[57] Zhuolin Jiang, Guangxiao Zhang, and Larry S. Davis. Submodular dictionary learning for sparse coding. In *CVPR*, pages 3418–3425, 2012.

[58] Zhuolin Jiang and Larry S. Davis. Submodular salient region detection. In *CVPR*, pages 2043–2050, 2013.

[59] Fan Zhu, Zhuoling Jiang, and Ling Shao. Submodular object recognition. In *CVPR*, 2014.

[60] Liangliang Cao, Zhenguo Li, Yadong Mu, and Shih-Fu Chang. Submodular video hashing: a unified framework towards video pooling and indexing. In *ACM Multimedia*, pages 299–308, 2012.

[61] Hanghang Tong, Jingrui He, Zhen Wen, Ravi Konuru, and Ching-Yung Lin. Diversified ranking on large graphs: an optimization viewpoint. In *KDD*, pages 1028–1036, 2011.

[62] Xiaojin Zhu, Andrew B. Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104, 2007.

[63] Jingrui He, Hanghang Tong, Qiaozhu Mei, and Boleslaw K. Szymanski. GenDeR: A generic diversified ranking algorithm. In *NIPS*, pages 1151–1159, 2012.

[64] Vidit Jain and Manik Varma. Learning to re-rank: query-dependent image re-ranking using click data. In *WWW*, pages 277–286, 2011.

[65] Winn Voravuthikunchai, Bruno Crémilleux, and Frédéric Jurie. Image reranking based on statistics of frequent patterns. In *ICMR*, page 129, 2014.

[66] Jun Yu, Yong Rui, and Dacheng Tao. Click prediction for web image reranking using multimodal sparse coding. *IEEE Transactions on Image Processing*, 23(5):2019–2032, 2014.

[67] Ricardo Omar Chávez, Hugo Jair Escalante, Luis Enrique Sucar, et al. Multimodal markov random field for image reranking based on relevance feedback. *ISRN Machine Vision*, 2013, 2013.

[68] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, pages 324–331, 2005.

[69] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28(4):1–38, 2010.

[70] Xiaohui Shen, Zhe Lin, Jonathan Brandt, Shai Avidan, and Ying Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, pages 3013–3020, 2012.

[71] Javed A. Aslam and Mark H. Montague. Models for metasearch. In *SIGIR*, pages 275–284, 2001.

[72] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.

[73] Raivo Kolde, Sven Laur, Priit Adler, and Jaak Vilo. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580, 2012.

[74] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993.

[75] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[76] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[77] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan. Visual classification with multitask joint sparse representation. *IEEE Transactions on Image Processing*, 21(10):4349–4360, 2012.

[78] Maksim Lapin, Bernt Schiele, and Matthias Hein. Scalable multitask representation learning for scene classification. In *CVPR*, 2014.

[79] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*. 2008.

[80] Liang Zheng, Liyue Sheng, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[81] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010.

[82] Dong Seon Cheng, Marco Cristani, Michele Stoppa, Loris Bazzani, and Vittorio Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011.

[83] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *ECCV*. 2014.

[84] Chunxiao Liu, Shaogang Gong, Chen Change Loy, and Xinggang Lin. Person re-identification: what features are important? In *ECCV*, 2012.

[85] Mert Dikmen, Emre Akbas, Thomas S Huang, and Narendra Ahuja. Pedestrian recognition with a learned metric. In *ACCV*, 2011.

[86] Martin Hirzer, Peter M Roth, Martin Köstinger, and Horst Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*. 2012.

[87] Zhen Li, Shiyu Chang, Feng Liang, Thomas S Huang, Liangliang Cao, and John R Smith. Learning locally-adaptive decision functions for person verification. In *CVPR*, 2013.

[88] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Re-identification by relative distance comparison. In *CVPR*, 2013.

[89] Martin Köstinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. *CVPR*, 2012.

[90] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*. 2014.

[91] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Learning midlevel filters for person reidentification. In *CVPR*, 2014.

[92] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.

[93] Alina Bialkowski, Simon Denman, Patrick Lucey, Sridha Sridharan, and Clinton B Fookes. A database for person re-identification in multi-camera surveillance networks. *DICTA*, 2012.

[94] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *ACM workshop*, 2011.

[95] Abir Das, Anirban Chakraborty, and Amit K Roy-Chowdhury. Consistent re-identification in a camera network. In *ECCV*. 2014.

[96] Brais Cancela, Timothy M Hospedales, and Shaogang Gong. Open-world person re-identification by multi-label assignment inference. 2014.

[97] Gang Wang and David A. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[98] Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[99] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *ECCV*, 2010.

[100] Dhruv Kumar Mahajan, Sundararajan Sellamanickam, and Vinod Nair. A joint learning framework for attribute models and object descriptions. In *ICCV*, 2011.

[101] Thomas Mensink, Jakob J. Verbeek, and Gabriela Csurka. Tree-structured CRF models for interactive image labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):476–489, 2013.

[102] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

[103] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Queen Mary. Person re-identification by attributes. In *BMVC*, 2012.

[104] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Towards person identification and re-identification with attributes. In *ECCV Workshops*, 2012.

[105] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Attributes-based re-identification. In *Person Re-Identification*, pages 93–117. Springer, 2014.

[106] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Re-id: Hunting attributes in the wild. In *BMVC*. 2014.

[107] Sheng-Jun Huang, Zhi-Hua Zhou, and ZH Zhou. Multi-label learning by exploiting label correlations locally. In *AAAI*, 2012.

[108] James Petterson and Tiberio S Caetano. Submodular multi-label learning. In *NIPS*, 2011.

[109] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *KDD*, 2010.

[110] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, 2011.

[111] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *KDD*, 2012.

[112] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

[113] Jianhui Chen, Lei Tang, Jun Liu, and Jieping Ye. A convex formulation for learning shared structures from multiple tasks. In *ICML*, 2009.

[114] Jianhui Chen, Ji Liu, and Jieping Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *KDD*, 2010.

[115] Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[116] Quanquan Gu, Zhenhui Li, and Jiawei Han. Learning a kernel for multi-task clustering. In *AAAI*, 2011.

[117] Paul Ruvolo and Eric Eaton. Online multi-task learning via sparse dictionary optimization. In *AAAI*, 2014.

[118] Lei Han, Yu Zhang, Guojie Song, and Kunqing Xie. Encoding tree sparsity in multi-task learning: A probabilistic framework. In *AAAI*, 2014.

[119] Lin Chen, Qiang Zhang, and Baoxin Li. Predicting multiple attributes via relative multi-task learning. In *CVPR*, pages 1027–1034, 2014.

[120] Sung Ju Hwang, Fei Sha, and Kristen Grauman. Sharing features between objects and their attributes. In *CVPR*, pages 1761–1768, 2011.

[121] Andy Jinhua Ma, Pong C. Yuen, and Jiawei Li. Domain transfer support vector ranking for person re-identification without target camera label information. In *ICCV*, 2013.

[122] Lianyang Ma, Xiaokang Yang, and Dacheng Tao. Person re-identification over camera networks using multi-task distance metric learning. *IEEE Transactions on Image Processing*, 23(8):3656–3670, 2014.

[123] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, pages 1042–1050, 2009.

[124] Linli Xu, Zhen Wang, Zefan Shen, Yubo Wang, and Enhong Chen. Learning low-rank label correlations for multi-label classification with missing labels. In *ICDM*, 2014.

[125] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, pages 365–372, 2009.

[126] Le An, Mehran Kafai, Songfan Yang, and Bir Bhanu. Reference-based person re-identification. In *AVSS*, pages 244–249, 2013.

[127] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Image Analysis*, pages 91–102. Springer, 2011.

[128] Doug Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *IEEE International Workshop on Performance Evaluation for Tracking and Surveillance*, 2007.

[129] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *ACM Multimedia*, pages 789–792, 2014.

[130] Mohammad Rastegari, Ali Farhadi, and David Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*. 2012.

[131] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Person re-identification by salience matching. In *ICCV*, 2013.

[132] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[133] Bryan Prosser, Wei-Shi Zheng, Shaogang Gong, Tao Xiang, and Queen Mary. Person re-identification by support vector ranking. In *BMVC*, 2010.

[134] Giuseppe Lisanti, Iacopo Masi, and Alberto Del Bimbo. Matching people across camera views using kernel canonical correlation analysis. In *ICDSC*, 2014.

[135] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, pages 2197–2206, 2015.

[136] Zhiyuan Shi, Timothy M Hospedales, and Tao Xiang. Transferring a semantic representation for person re-identification and search. In *CVPR*, pages 4184–4193, 2015.

[137] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[138] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

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

[140] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. 2014.

[141] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2015.

[142] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[143] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361. 2014.

[144] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2155–2162, 2014.

[145] Yuting Zhang, Kihyuk Sohn, Ruben Villegas, Gang Pan, and Honglak Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In *CVPR*, pages 249–258, 2015.

[146] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *NIPS*, pages 2553–2561, 2013.

[147] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *CVPR*, pages 4703–4711, 2015.

[148] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen Change Loy, and Xiaoou Tang. Deepid-net: Deformable deep convolutional neural networks for object detection. *CoRR*, abs/1412.5661, 2014.

[149] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[150] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*.

[151] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.

[152] Piotr Dollár, Ron Appel, and Wolf Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, pages 645–659, 2012.

[153] Markus Mathias, Rodrigo Benenson, Radu Timofte, and Luc J. Van Gool. Handling occlusions with franken-classifiers. In *ICCV*, pages 1505–1512, 2013.

[154] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, pages 3476–3483, 2013.

[155] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *CVPR*, pages 5325–5334, 2015.

[156] Anelia Angelova, Alex Krizhevsky, Vincent Vanhoucke, Abhijit Ogale, and Dave Ferguson. Real-time pedestrian detection with deep network cascades. In *BMVC*, 2015.

[157] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015.

[158] Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, pages 1798–1807, 2015.

[159] Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. DeepProposal: Hunting objects by cascading deep convolutional layers. In *ICCV*, 2015.

[160] Nikolaos Karianakis, Thomas J. Fuchs, and Stefano Soatto. Boosting convolutional features for robust object proposals. *CoRR*, abs/1503.06350, 2015.

[161] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, pages 4380–4389, 2015.

[162] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[163] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[164] Pedro F. Felzenszwalb, Ross B. Girshick, and David A. McAllester. Cascade object detection with deformable part models. In *CVPR*, pages 2241–2248, 2010.

[165] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1532–1545, 2014.

[166] Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *CVPR*, pages 4749–4757, 2015.

[167] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. *CoRR*, abs/1406.5774, 2014.

[168] Joe Yue-Hei Ng, Fan Yang, and Larry S. Davis. Exploiting local features from deep networks for image retrieval. *CoRR*, abs/1504.05133, 2015.

[169] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[170] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *CVPR*, pages 1903–1911, 2015.

[171] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, pages 1–42, April 2015.

[172] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405. 2014.

[173] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):2071–2084, 2015.

[174] Bojan Pepik, Michael Stark, Peter V. Gehler, and Bernt Schiele. Multi-view and 3d deformable part models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(11):2232–2245, 2015.

[175] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Learning to detect vehicles by clustering appearance patterns. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2511–2521, 2015.

[176] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *ICCV*, 2015.

[177] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning strong parts for pedestrian detection. In *ICCV*, 2015.