IMPROVING SET-BASED FACE RECOGNITION

A Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Mengjun Leng

May 2019

IMPROVING SET-BASED FACE RECOGNITION

Mengjun Leng

APPROVED:

Ioannis A. Kakadiaris, Chairman
Dept. of Computer Science

Guoning Chen
Dept. of Computer Science

Thamar Solorio
Dept. of Computer Science

Zhu Han
Dept. of Electrical and Computer Engineer

Dean, College of Natural Sciences and Mathematics

ACKNOWLEDGMENTS

Many people have helped me during my Ph.D. program. More specifically, I would like to express my appreciation to my advisor, my dissertation committee members, my colleagues, my family, and my friends. Without them, I would not have finished this dissertation.

Firstly, I am indebted to my advisor, Prof. Ioannis A. Kakadiaris, who offered me the great opportunity of pursuing my Ph.D. in the Computational Biomedicine Laboratory (CBL). I appreciate this opportunity. He has not only served as my dissertation advisor, but as a life-long mentor. Since my first meeting with him, I have been impressed by his enthusiasm towards research, life, and the people around him. While working with him, I have learned to take the initiative, to manage my time, and to work hard. On the one hand, he is every critical of our research and professional skills, yet on the other hand, he is also very supportive. There have been many tough moments during these past five years, and without Prof. Kakadiaris' support, I would have quit many times.

Besides my advisor, I would like to thank the rest of my dissertation committee members, namely Prof. Guoning Chen, Prof. Thamar Solorio, and Prof. Zhu Han, for their great support and invaluable advice. Their broad knowledge in data visualization, graphics, natural language processing, and wireless networking, have been a great inspiration to me. I am genuinely proud to have them as committee members for this dissertation.

I want to thank all my previous and current colleagues at CBL. During these five years, we have learned and made progress together. I enjoyed working with them. Special thanks to Dr. Panagiotis Moutafis for guiding me into the research world at the very early stage

of my Ph.D. I want to thank Dr. Pengfei Dou and Dr. Lingfeng Zhang for sharing a lot of their experience. I am grateful to Dr. Yuhang Wu. We joined CBL in the same period and worked in the same room over the last two semesters. He is a very tough learner and makes progress every day. Thanks to Mr. Nikolaos Sarafianos and Mr. Lei Shi; I feel happy to see them every day in the lab. Thanks to Mr. Xiang Xu and Mr. Le Anh Vu Ha; they contributed a lot to the pipeline, which facilitates the work of everyone in the lab. I also appreciate the advice given by Dr. Ioannis Konstantinidis. Thanks to Mr. Charles Livermore, Mr. Ali Memariani, Mr. Christos Smailis, Dr. Dimitrios Spiliotopoulos, Dr. Tian Xie, and Dr. Michalis Vrigkas. It was nice working with them.

I appreciate the support and understanding of my family, especially my parents. I have been in the United States for more than five years. Having had to take care of all my daily needs has helped me to understand better how much my parents did for me before I left home. I am now grown up, and they are getting old; I feel very sorry that I can not take good care of them by their side.

I would like to thank my friends for supporting me, both physically and mentally. Special thanks go to Dr. Yuanyuan Zhao and Dr. Xiaoxi Zhang, for their help and encouragement, both in my life and in my research.

Thanks are also due to the U.S. Department of Homeland Security, under Grant Award Number 2015-ST-061-BSH001 and 2017-ST-BTI-00001-02-01, the US Army Research Lab, under Grant Award Number W911NF-13-1-0127, as well as the University of Houston, for their financial support. I am also grateful for the support of the Core Facility for Advanced Computing and Data Science at the University of Houston for assisting me with the calculations that were required for this dissertation.

IMPROVING SET-BASED FACE RECOGNITION

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Mengjun Leng

May 2019

Abstract

A face image set is a group of face images from the same person. In set-based face recognition systems, face image sets are employed either in the gallery, probe, or both for comparisons. Compared with a single face image, an image set provides more information; hence, better performance is expected. However, it also brings a lot of challenges and remains an open problem in real-life scenarios. First, there are large variations within an image set (e.g., poses, expressions, and occlusions). Second, the number of images varies for different image sets. Third, there may be outliers in a set due to misdetection or mistracking. Fourth, the computation and storage costs are very high, especially for large-scale image sets. The goal of this dissertation is to design effective and efficient algorithms in template generation and matching that can represent identity information and take advantage of the within-set variations. The first contribution is a set-based prototype and metric learning algorithm (SPML) that generates compact templates and robust similarity measurements for set-to-set matching. The second contribution is a confidence-driven network (CDN) to quantify the confidence level of images in a set and enhance the point-to-set matching. The third contribution is a confidence prediction network (CPN) that can serve as an add-on module to enhance the performance of a sample-based face recognition system for set-based face recognition tasks. The fourth contribution is an attention-based recursive binary embedding (ARBE) algorithm to extract binary templates for face image sets. The proposed algorithms achieved significant improvements when compared with previous advances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Face-recognition technology has been adopted in many identification systems to distinguish identity in an easier, faster, and more secure manner. To integrate more information to enhance the recognition performance, image sets are employed in the set-based face-recognition system. An image set contains a group of images describing the same individual. In set-based face recognition systems, there are mainly two tasks: (i) the set-to-set matching, where comparisons are conducted between two face-image sets, and (ii) the point-to-set matching, where a single face image and a face image set are compared. Many of the face data acquired in real-life are face-image sets by nature. For example, face frames extracted via face tracking from video streams are groups of face images from the same person. In a multi-camera setting, face images captured by multiple cameras also form an image set. As a result, many real-life applications can be mapped into the problem

of set-based matching. When tracking a person across a video camera network, it is typically a matching between tracking results from different cameras (i.e. set-to-set matching). When the police locate a missing child, it is usually a matching between an ID photo with the tracking results from each video camera (i.e. point-to-set matching). Compared with single-image-based matching, an image set can provide more information. Hence better recognition performance is expected.

1.2 Challenges

A common approach to compare two face images is first to generate a template (usually a feature vector) for each image, and second, compute the similarity between the generated templates. To adapt the same strategy to set-based matching, first templates need to be created for face-image sets, and second, the similarity between the generated templates need to be computed. Except for the common challenges in sample-based face-recognition system (e.g., poses and occlusions), the image set settings also bring several new challenges.

1. **The within-set variations could be large:** Since there are multiple images in an image set, these images may exhibit differences in poses, expressions, and face sizes. As a result, the generated template should either be robust to all variations or represent all variations.

2. **The number of images varies:** The numbers of images in an image set are usually different. It requires the capability to process variable numbers of images in template generation.

3. **Image sets may contain outliers:** The outliers in an image set are images that do not contain faces of the labeled identity. It could be faces from other identities or non-face images via misdetection or mistracking. Outliers will degrade the matching performance.

4. **The problem complexity is increased:** From a single image to an image set, the computational complexity is increased for template generation, similarity measurements or both. Moreover, most of the datasets for face recognition are organized for sample-based face recognition. Extra effort is needed to construct protocols for both training and evaluation.

1.3 Limitations of Previous Work

Existing methods address some of the challenges mentioned in Section 1.2. Besides, they are limited in at least one of the following aspects.

1. **Information redundancy:** Existing approaches usually take all images in an image set to build the representation model. This model contains redundant information that is duplicated or useless for the recognition. Additionally, this redundancy is time and storage consuming.

2. **Non-compact template:** The set representation is not compact enough. Most of the templates generated for image sets are in high-dimensional feature space or contain very complex structures.

3. **Neglect the within set variations:** Some of the methods treat all the images in the set equally. Within an image set, different images contain various levels of discriminative information and thus should contribute differently to the results.

4. **Complex model assumptions:** Most of the methods rely on strong model assumptions (i.e. sub-space model, hull-based model, or statistical model), which do not necessarily hold. Moreover, the model estimation is computationally expensive, especially in a high-dimensional feature or sample space.

1.4 Goal and Objectives

The goal of this dissertation is to develop effective and efficient algorithms to achieve statistically significant improvements in matching for set-based face recognition. In this dissertation, new algorithms are proposed to address the challenges and limitations highlighted in Section 1.2 and 1.3. In particular, the objectives are to:

1. Develop and evaluate an algorithm that generates compact templates and robust similarity measurements for set-to-set matching.

2. Develop and evaluate algorithms that generate common templates for point-to-set matching.

3. Develop and evaluate an add-on module that can enhance the sample-based face-recognition system for set-to-set and point-to-set matching.

4. Develop and evaluate a compact binary template for face image sets.

In particular, objective 1 targets on challenge 1-3 and limitation 1-3; objective 2 targets on challenge 1-2 and limitation 3-4; objective 3 targets on challenge 1-2, 4 and limitation 2-4; objective 4 targets on challenge 1-2 and limitation 2-4.

1.5 Contributions to Date

1.5.1 Objective 1

The set-based prototype and metric-learning framework (SPML) is proposed for set-to-set matching. In particular, each gallery image set is represented using a reduced-affine hull spanned by a few learned prototypes. The affine-hull model is employed, which can (i) preserve all the within set variations (challenge 1 and limitation 3), and (ii) accept variable numbers of images in a set (challenge 2). Instead of using all the images in a set, a reduced number of prototypes are learned, which (i) reduces the template size and information redundancy (limitation 1 and 2), and (ii) is more robust to outliers (challenge 2). To maintain or even improve the recognition performance, a Mahalanobis metric is learned simultaneously with the prototypes. The optimization problem is formulated using a single loss function that jointly learns the prototypes and metric to bring similar image sets closer to each other, while separating dissimilar ones. The principal contribution of this work is a framework with the following advantages:

1. It uses fewer prototypes to represent each image set in the gallery, reducing the computational cost and storage requirement.

2. It increases the robustness of the hull model.

3. It can be used with any hull model and any distance metric learning objective function.

1.5.2 Objective 2

The confidence driven network (CDN) is proposed to generate templates for point-to-set matching. The proposed template contains two parts: (i) the regular feature representation, and (ii) a confidence score to measure the discriminative information level of first part of the template. The confidence score can help differentiate the contribution of each image in a set. In other words, the within-set variations are encoded into the confidence scores (challenge 1 and limitation 3). In the training phase, the confidence score will guide the feature extraction network to focus less attention on the samples with lower confidence levels. By doing so, it avoids overfitting on samples that are possibly difficult, or others that the model is uncertain about its predictions. In the matching phase, the confidence scores are used to integrate the results from different samples of the same set via simple weighted average fusion (challenge 2 and limitation 4). The principal contributions of this work are the following:

1. A weighted-by-confidence point-to-set triplet loss that enables us to adapt a point-to-point network to a point-to-set network,

2. A single-sample test mechanism to quantify the discriminative level of a sample.

1.5.3 Objective 3

CDN is extended into a confidence prediction network (CPN), which acts as an add-on module to adapt a sample-based face recognition (FR) system for set-based FR applications. CPN can be used to generate confidence scores as attention and aggregate information from different images into a single template. As a result, it (i) is free from model assumptions (limitation 4), and (ii) reduces the template size (limitation 2). Similar to CDN, the within-set variations are encoded into the confidence scores (challenge 1 and limitation 3). In particular, the single-sample-test mechanism is extended to generate a global pseudo-ground-truth for the confidence score such that the confidence scores can be learned: (i) independently without the access to the template of a sample-based FR system, and (ii) without set-based restriction in datasets (challenge 4). The proposed CPN is an add-on module with the following advantages:

1. The training of feature representations and the confidence score are completely independent which simplify the training process a lot.

2. CPN can work with different face recognition systems and enhance the performance without changing the systems.

1.5.4 Objective 4

The attention-based recursive binary embedding (ARBE) algorithm is proposed to extract the binary template for image sets. Specifically, the network contains two parts: (i) attention-based feature learning, and (ii) recursive binary coding. In the first part,

following CDN, a real-valued feature representation is learned for each sample with a corresponding attention score. The attention score describes the contribution of the corresponding sample representation (challenge 1-2, limitation 3-4). In the second part, each bit is learned recursively. The output of the previous bit is used as meta input when learning the current bit. The results from different samples are integrated at each bit (limitation 2). The proposed ARBE results in an increased recognition performance compared to the sequential code, while the number of projections is still restricted to a linear relation to code length. Learning from the recent advances in face recognition and image set classification, the angular-based similarity and the image set attention schemes are also adapted in the framework. The contribution is a new binary-embedding framework for a face image set with the following advantages:

1. It increases the recognition power while maintaining a linear model complexity.
2. It is designed under a standard neural network architecture so that it can be easily integrated into different network design.

1.6 Dissertation Outline

The rest of the dissertation is organized as follows: the background and related work are presented in Chapter 2. The proposed methods for each of the objectives are discussed and evaluated in Chapter 3 to Chapter 6, respectively. Finally, Chapter 7 concludes all the works and provides directions for future research.

1.7 Publications

1. **M. Leng**, and I.A. Kakadiaris, Confidence prediction network for face image sets: A plug-and-play approach, In *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019 (under review).

2. **M. Leng**, and I.A. Kakadiaris, Recursive binary template embedding for face image sets, In *Proc. International Conference on Biometrics: Theory, Applications and Systems*, Los Angeles, CA, 2018.

3. **M. Leng**, and I.A. Kakadiaris, Confidence-driven network for point-to-set matching: Application to multi-probe face identification, In *Proc. International Conference on Pattern Recognition*, Beijing, China, 2018.

4. **M. Leng**, P. Moutafis, and I.A. Kakadiaris, Joint prototype and metric learning for image set classification: Application to video face identification, In *Image and Vision Computing*, 58:204-213, 2017.

5. **P. Moutafis**, M. Leng, and I.A. Kakadiaris, Regression-based Metric Learning, In *Proc. International Conference on Pattern Recognition*, Cancún, Mexico, Dec. 4-8, 2016.

6. **P. Moutafis**, M. Leng, and I.A. Kakadiaris, An overview and empirical comparison of distance metric learning methods, In *IEEE Transactions on Cybernetics*, 47(3): 612-625, 2016.

7. **M. Leng**, P. Moutafis, and I.A. Kakadiaris, Joint prototype and metric learning for set-to-set matching: Application to biometrics, In *Proc. International Conference on Biometrics: Theory, Applications and Systems*, Arlington, VA, 2015.

Chapter 2

Background and Related Work

In this chapter, an overview of essential concepts and existing literature in set-based matching and other related fields is offered. A general pipeline for the set-based matching can be illustrated in Figure 2.1.

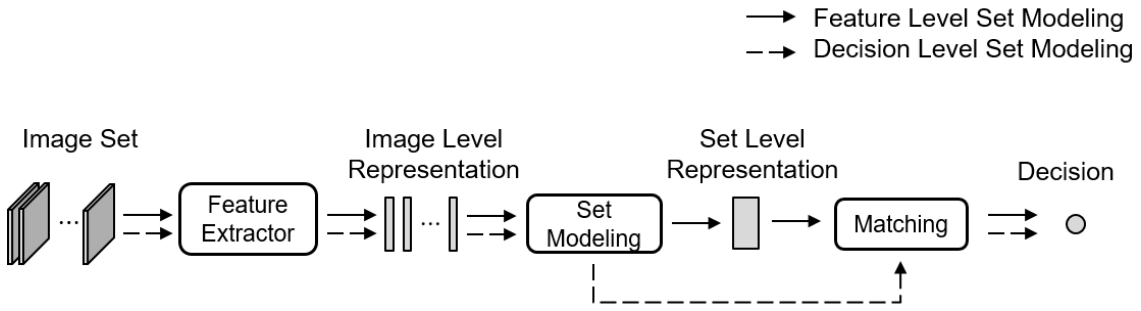

Figure 2.1: Illustration of a general set-based matching pipeline. Solid lines depict the feature-level set modeling flow. Dash lines depict the decision-level set modeling flow.

It can be viewed from a fusion perspective. According to the step where the fusion happens, methods can be grouped into two categories: (i) feature-level set modeling, and (ii) decision-level set

modeling. In feature-level modeling methods (i.e. following solid arrows in Figure 2.1), the image-level feature representations are embedded into a single set-based representation. In decision-level modeling methods, the image-level representations are taken directly into the matcher and return a single decision. In both categories, there are two key issues to solve: (i) how to model an image set, and (ii) how to conduct discriminative learning. To this end, a brief introduction to the image set modeling is introduced and then the discriminant analysis under the corresponding models is reviewed. Expect for the set-based matching, advances in the field of performance prediction (objective 2 and 3) and binary templates (objective 4) are also involved in this thesis. Related literature in these two fields is also discussed in this chapter.

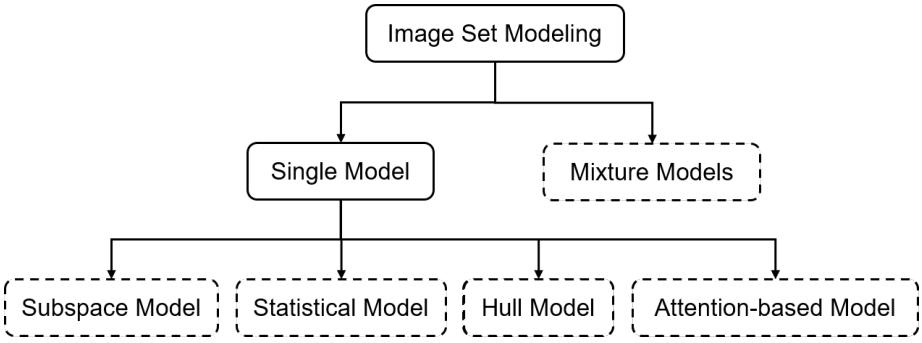

Figure 2.2: Illustration of the taxonomy tree for image set modeling.

2.1 Image Set Modeling

In this section, the collected papers are clustered according to how they model an image set to structure the existing literature. Specifically, a taxonomy tree is proposed in Figure 2.2. The image set modeling can be divided into two groups: (i) single model, and (ii) mixture model. In the first group, algorithms model each image set using a single model, while in the second group they model

each image set using a mixture of different models. Specifically, in the single model approaches, four kinds of model are used: (i) subspace model, (ii) statistical model, (iii) hull model, and (iv) attention-based model. To give a structured description, the collected literature is clustered into the leaf categories: the subspace model (Table 2.1), the statistical model (Table 2.2), the hull model (Table 2.3), the attention-based model (Table 2.4), and the mixture model (Table 2.6 and Table 2.5). Each of the categories will be discussed in depth in the following subsections, including (i) the basic modeling framework, (ii) variations in different approaches, and (iii) advantages and limitations. Except for the different set modelings, the paper highlight and fusion level for each algorithm are also listed. Specifically, the contributions of each algorithm are summarized in the paper highlight column. In the fusion level column, the feature fusion and decision fusion are distinguished.

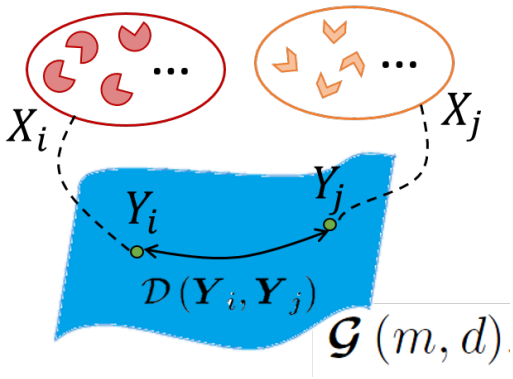

Figure 2.3: Illustration of subspace model. $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_j$ denote two linear subspaces lying on the Grassmann Manifold $\mathcal{G}(m, d)$. $\mathcal{D}(\boldsymbol{Y}_i, \boldsymbol{Y}_j)$ denotes the geodesic distance on the manifold that is the length of the shortest path on the surface. It is a natural dissimilarity measurement of the image set $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$.

2.1.1 Subspace Model

An image set $\boldsymbol{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ can be modeled using an m dimensional linear subspace of $\mathbb{R}^d$. The linear subspace can be specified by $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_m] \in \mathbb{R}^{d \times m}$, an orthogonal matrix contains m base vectors. This linear subspace model can be constructed using singular value decomposition, where $\boldsymbol{Y}$ contains the eigenvectors corresponding to the m largest eigenvalues [22]. All the m dimensional linear subspaces of $\mathbb{R}^d$ form a Grassmann manifold $\boldsymbol{\mathcal{G}}(m, d)$. As a result, the subspaces extracted from original image sets can be treated as different points on a manifold. A straightforward dissimilarity measurement is the geodesic distance on the manifold, as illustrated in Figure 2.3. However, it suffers in two aspects. First, it is computationally very expensive. Second, this distance lies on the Grassmann manifold. Discriminant analysis techniques that developed in Euclidean space cannot be directly used. Instead, different kernel metrics are developed to embed the Grassmann manifold to a Reproducing Kernel Hilbert Space (RKHS). The most commonly used ones are the projection kernel and the Binet-Cauchy kernel proposed in [22]. The projection metric is defined as

$$\boldsymbol{\mathcal{K}}_P(\boldsymbol{Y}_i, \boldsymbol{Y}_j) = \|\boldsymbol{Y}_i^T \boldsymbol{Y}_j\|_F, \tag{2.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. It can help to understand by associating the isometric embedding [11],

$$\psi_P : \boldsymbol{\mathcal{G}}(m, d) \mapsto \mathbb{R}^{d \times d}, \boldsymbol{Y}_i \mapsto \boldsymbol{Y}_i^T \boldsymbol{Y}_i. \tag{2.2}$$

The projection kernel is simply the Euclidean distance in $\mathbb{R}^{d \times d}$. Similarly, the Binet-Cauchy kernel is defined as

$$\boldsymbol{\mathcal{K}}_{BC}(\boldsymbol{Y}_i, \boldsymbol{Y}_j) = \det\left(\boldsymbol{Y}_i^T \boldsymbol{Y}_j \boldsymbol{Y}_j^T \boldsymbol{Y}_i\right). \tag{2.3}$$

Using the same embedding, a set of more general kernels is proposed in [27], including the polynomial kernel, RBF kernel, Laplace kernel, binomial kernel and logarithm kernel. The kernels

Table 2.1: Papers using the subspace model.

Abbr.	Paper Highlights	Fusion level
PML [38]	Proposes a geometry-aware dimension reduction directly from the original Grassmann Manifold to a more discriminative lower dimensional Grassmann Manifold.	Feature
$k_{.,p}$/ $k_{.,bc}$ [27]	Introduces a group of positive definite kernels (including universal ones) to embed Grassmannians into Hilbert space via Plücker embedding or projection embedding.	Decision
GDL/ KGDL [26]	Proposes a sparse coding and dictionary learning framework on Grassmann manifold via embedding it into the space of symmetric matrices. Devises a close form solution for dictionary learning and can be kernelized.	Feature
GGDA [28]	Proposes a discriminant analysis algorithm that preserves both manifold structure and local structure of the data based on graph embedding framework [84].	Feature
GFKS2V [91]	Consider point-to-set matching as heterogeneous subspaces lying on Grassmann manifold and use Geodesic Flow Kernel to build the connection.	Decision

mentioned above enable us to leverage the subspace model with the techniques developed in Euclidean spaces (e.g., support vector machines [66]) for a more discriminative classification. In particular, a kernel-based graph embedding framework is proposed in [28] to preserve the local structure. The sparse coding framework was developed in [26] with an unsupervised dictionary learning strategy. It was further extended into a general Riemannian coding framework [24] with both supervised and unsupervised dictionary learning strategies. Instead of first embedding the Grassmann manifold into a Hilbert space, the projection metric learning [38] was proposed to conduct the dimension reduction directly from a manifold to a lower dimensional manifold to preserve

essential manifold structures. These discriminant analysis techniques will be discussed in Section 2.2. There are some other algorithms employing multiple linear subspace models in different local areas [72, 10]. They are grouped in the mixture model and will be discussed in Section 2.1.5.

In summary, the subspace-based approaches assume that an image set lies on a low dimensional subspace. It can take advantage of the well-developed Riemannian geometry to leverage rich discriminant analysis techniques in the Euclidean space. It provides a loose characterization of the set variations. On the one hand, this is robust to noise. On the other hand, it discards the variations in different directions, which may result in losing important local information. Additionally, the dimension of the linear subspace is crucial for the performance and needs to be tuned.

2.1.2 Statistical Model

Approaches in this category attempt to characterize each image set use either its statistical properties [24, 39, 25, 66, 71] or the probabilistic distribution [52].

Statistical Property: Using the statistical properties is a straightforward way to structure the image-set data. The most frequently used are the first order statistic (i.e. mean value) and the second order statistic (i.e. covariance matrix). The first order statistic lies in the Euclidean space, and thus the Euclidean distance can be directly used as the dissimilarity measurement. The second order statistic $\boldsymbol{C}$ of an image set $\boldsymbol{X}$ is defined as

$$\boldsymbol{C} = \frac{1}{N-1} \sum_{i=1}^{N} \left(\mathbf{x}_i - \boldsymbol{\mu}\right) \left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T, \tag{2.4}$$

where $\boldsymbol{\mu}$ is the mean vector of all samples in $\boldsymbol{X}$. The non-singular covariance matrix lies on the Riemannian manifold spanned by the $d \times d$ SPD matrices [56].

There are several well explored metrics in the Riemannian geometry. The first is affine-invariant

Table 2.2: Papers using the statistical model.

Abbr.	Paper Highlights	Fusion level
LEML [39]	Employs the covariance model and learns a projection directly in the tangent space at identity matrix, which reserves the symmetric property of the original SPD tangent map.	Feature
SSCIS [52]	Proposes to cluster the whole training data and compute the discrete distribution of each class across different clusters. The modified Bhattacharyya distance between distributions is used to measure the similarity.	Feature
SPD-ML [25]	Proposes to use orthonormal projection for SPD dimension reduction.	Feature
CDL [71]	Models each image set using the covariance matrix and extends LDA and PLS to the Log-Euclidean distance.	Feature
LERM [37]	Proposes a hierarchical cross domain learning between point data on Euclidean space and set data on Riemannian Manifold.	Feature
DCLR [77]	An end-to-end framework for set-based matching focusing on a covariance based loss.	Feature
SPDNet [35]	Proposes the first deep network that takes SPD as input.	Feature

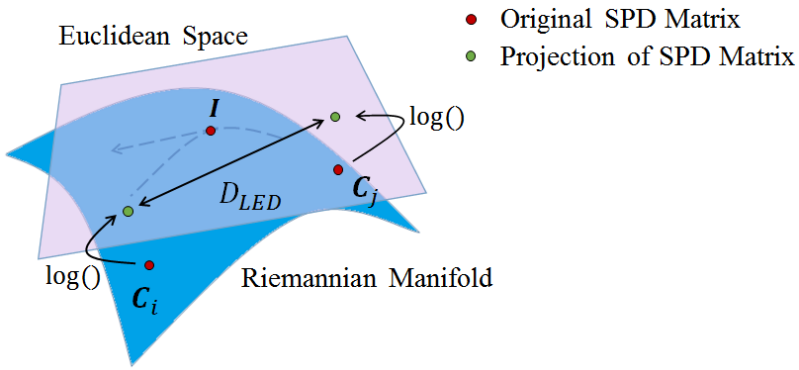

Figure 2.4: Illustration of Log-Euclidean distance. $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ denote two covariance matrices on the Riemannian Manifold. I is the identity matrix. The Log-Euclidean distance is equivalent to first project the SPD matrices on Riemannian Manifold onto a Euclidean space via logarithm transform.

distance (AID) [58]. The AID between two covariance matrices $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ is defined as

$$\mathcal{D}_A\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) = \sqrt{\sum_{t=1}^{d} ln^2 \lambda_t(\boldsymbol{C}_i, \boldsymbol{C}_j)}, \tag{2.5}$$

where $\lambda_t(\boldsymbol{C}_i, \boldsymbol{C}_j)(t = 1, ..., d)$ are obtained from $|\lambda \boldsymbol{C}_i - \boldsymbol{C}_j| = 0$. However, the AID is computationally very expensive. The Stein metric [62] and Log-Euclidean distance (LED) [3] show several similarities to the AID while being less expensive to compute. The stain metric on SPD manifolds is defined as

$$\mathcal{D}_S\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) = ln\ \det\left(\frac{\boldsymbol{C}_i + \boldsymbol{C}_j}{2}\right) - \frac{1}{2} ln\ \det\left(\boldsymbol{C}_i \boldsymbol{C}_j\right) \tag{2.6}$$

LED is the most commonly used one, and is defined as

$$\mathcal{D}_L\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) = \left\| log(\boldsymbol{C}_i) - log(\boldsymbol{C}_j) \right\|_F, \tag{2.7}$$

where $log()$ is the ordinary matrix logarithm operation and $\|\cdot\|_F$ denotes the Frobenius norm. The LED metric is illustrated in Figure 2.4. It can be considered as projecting a point $\boldsymbol{C}_i$ on the Riemannian manifold to a Euclidean space using the logarithm map. The projected Euclidean space

is a tangent space at the point of identity matrix $\boldsymbol{I}$. The covariance matrix was first proposed to structure an image set in [71] with the LED. Linear Discriminant Analysis (LDA) and Partial Least Squares regression (PLS) are extended to the Log-Euclidean distance using kernel trick for a more discriminative projection. The kernel learning framework [66] and the general Riemannian coding framework [24] are also applied to the covariance-based image set classification. To reduce the computational cost, SPD Manifold learning (SPD-ML) proposes to conduct manifold-to-manifold dimension reduction with orthogonal projection. Log-Euclidean metric learning (LEML) [39] achieves the same goal via collaborating metric learning with a tangent map. There are other algorithms [51, 36] employing multiple statistical properties. They are grouped in the mixture model and will be discussed in Section 2.1.5.

In summary, approaches that rely on the statistical properties do not hold any assumption on the structure of the image set. As a result, they are more robust to different data distributions and can be applied to a broader scenario. To get a reliable estimation of statistical property, enough samples are necessary. As a result, they are more suitable for large-scale image sets.

Probabilistic Distribution: Besides the statistical properties, image sets can also be structured using their probability density distributions. The distribution-based distances for the general purpose, like Kullback-Leibler divergence (KLD),

$$\mathcal{D}_K\left(\mathcal{P}_i, \mathcal{P}_j\right) = \sum \mathcal{P}_i(\mathbf{x}) log \frac{\mathcal{P}_i(\mathbf{x})}{\mathcal{P}_j(\mathbf{x})}, \tag{2.8}$$

and the Bhattacharyya Distance,

$$\mathcal{D}_B\left(\mathcal{P}_i, \mathcal{P}_j\right) = -log\left(\sqrt{\sum \mathcal{P}_i(\mathbf{x})\mathcal{P}_j(\mathbf{x})}\right), \tag{2.9}$$

are a nature dissimilarity measurement. These two distances can be applied for any kind of distribution. Specifically for the Gaussian distribution, there is another less trivial approach. According to the information geometry, a d dimensional Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{C}\right)$ can be embedded

onto a $d+1$ dimensional Riemannian Manifold via the following mapping [36],

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C}) \mapsto |\boldsymbol{Q}|^{-\frac{2}{d+1}} \begin{bmatrix} \boldsymbol{Q}\boldsymbol{Q}^T & \boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{\mu} \\ \boldsymbol{\mu}^T & \mathbf{1} \end{bmatrix}. \tag{2.10}$$

where $\boldsymbol{Q}$ is the Cholesky decomposition of $\boldsymbol{C}$. As a result, the AID (Equation (2.5)) and LED (Equation (2.7)) can be applied to the embedded Gaussian distribution directly. In the existing literature, most algorithms assume Gaussian distribution [61, 36] or a mixture of Gaussian [75]. Other approaches take the histogram across some defined bins as the distribution. The semi-supervised spectral clustering algorithm [29] proposes to first cluster all the images into a certain number of clusters. For each image set, the probability of its samples distribution across different clusters is used as the set model.

In summary, the probabilistic distribution-based approaches hold strong assumptions concerning the distribution of the data, which may not always be true. For the general purpose distribution-based distance, very limited discriminative analysis techniques are proposed. The embedding approach for the Gaussian distribution can take advantage of the well-developed Riemannian geometry and is expected to provide more discriminant measurement.

2.1.3 Hull Model

Approaches in this category attempt to represent each image set using an affine hull [8] or other type of reduced affine hull. The image set $\boldsymbol{X}$ can be represented using a hull spanned by all the samples, using the general formula

$$\begin{aligned} \mathcal{H}(\boldsymbol{X}) &= \{\boldsymbol{\alpha}\boldsymbol{X} | \alpha_n \in \boldsymbol{\mathcal{R}}\} \\ &= \left\{ \sum_{n=1}^{N} \alpha_n \mathbf{x}_n \middle| \sum_{n=1}^{N} \alpha_n = 1, \alpha_n \in \boldsymbol{\mathcal{R}} \right\}. \end{aligned} \tag{2.11}$$

$$\begin{aligned} \mathcal{D}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) &= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} ||\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j||_2^2 \\ \text{s.t. } \sum_n \alpha_i^n = 1,\ &\sum_n \alpha_j^n = 1,\ \alpha_i^n, \alpha_j^n \in \boldsymbol{\mathcal{R}}. \end{aligned} \tag{2.12}$$

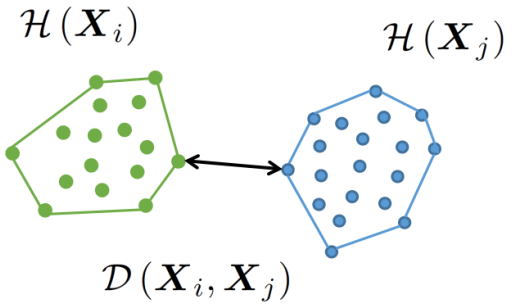

Figure 2.5: Illustration of hull-based model. $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ denote two hulls spanned by two image set $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ respectively. $\mathcal{D}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right)$ denotes the geometric distance between two hulls.

A hull contains the linear combinations of all samples in $\boldsymbol{X}$, with a common restriction ($\sum_n \alpha_n = 1$), and some defined restrictions ($\boldsymbol{\mathcal{R}}$) on the combination coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_{N_i}]^T$. Under this model, hull-to-hull distance is used as the dissimilar measurements between image sets. It is defined as the Euclidean distance between the closest points (i.e. Figure 2.5) on the two hulls. It can be observed that all the samples $\mathbf{x}_i^n$ only appear in quadratic terms. The distance in Equation (2.12) can be kernelized as

$$\begin{aligned} \mathcal{D}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j}\ & \boldsymbol{\alpha}_i^T\boldsymbol{\mathcal{K}}_{ii}\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_i^T\boldsymbol{\mathcal{K}}_{ij}\boldsymbol{\alpha}_j \\ & - \boldsymbol{\alpha}_j^T\boldsymbol{\mathcal{K}}_{ij}^T\boldsymbol{\alpha}_j + \boldsymbol{\alpha}_j^T\boldsymbol{\mathcal{K}}_{jj}\boldsymbol{\alpha}_j \\ \text{s.t. } \sum_n \alpha_i^n = 1,\ &\sum_n \alpha_j^n = 1,\ \alpha_i^n, \alpha_j^n \in \boldsymbol{\mathcal{R}}, \end{aligned} \tag{2.13}$$

where $\boldsymbol{\mathcal{K}}_{ij}$ is the kernel matrix corresponding to $\boldsymbol{X}_i^T\boldsymbol{X}_j$. The corresponding point-to-set distance

between an image set $\boldsymbol{X}_i$ and a single image $\boldsymbol{p}_j$ is defined as

$$\begin{aligned} &\mathcal{D}\left(\boldsymbol{X}_i, \boldsymbol{p}_j\right) = \min_{\boldsymbol{\alpha}_i} ||\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{p}_j||_2^2 \\ &\text{s.t. } \sum_n \alpha_i^n = 1,\ \alpha_i^n \in \boldsymbol{\mathcal{R}}. \end{aligned} \tag{2.14}$$

As the hull-to-hull and point-to-hull distance are defined on the closest points, the specific restriction $\boldsymbol{\mathcal{R}}$ is crucial. An under-restricted hull model is sensitive to outliers, which results in the overlap of inter-class hulls. However, an over-restricted hull model is less tolerant of within-set variations, which results in losing local information. To find the balance, different hull models are designed with different restrictions. The affine hull model and convex hull model are first proposed in [8]. In an affine hull, there is no defined restriction $\boldsymbol{\mathcal{R}} = (-\infty, +\infty)$. In a convex hull, $\boldsymbol{\mathcal{R}} = [0, U],\ U \geq 1$. Since the convex hull is usually over-restricted and the affine hull model is sensitive to outliers, they also proposed intermediate models with $\boldsymbol{\mathcal{R}} = [L, U],\ L < 0,\ U \geq 1$. The affine hull restriction is employed in [34]. However, additional sparse constraints are added when searching for the nearest points on affine hulls:

$$\begin{aligned} \mathcal{D}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} &||\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j||_2^2 \\ &+ \lambda_1\|\boldsymbol{\alpha}_i\|_1 + \lambda_2\|\boldsymbol{\alpha}_j\|_1 \\ \text{s.t. } \sum_n \alpha_i^n = 1,\ &\sum_n \alpha_j^n = 1, \end{aligned} \tag{2.15}$$

where $\|\cdot\|_1$ denotes the l_1 norm. The l_1 norm regularization makes distance computation time consuming. Instead, the regularized affine hull model (RAH) [86] was proposed with l_2 constraint. They also provide a fast-solver for the l_2 norm constraint. The RAH is proved to be both effective and efficient. As a result, it is employed in the supervised set-to-set distance metric learning (SSDML) [89]. A Gaussian distribution-based constraint is proposed in [76]. When searching for the closest points on two hulls, it maximizes the probability of both samples belonging to the corresponding affine hull to restrict the impact of outliers. These algorithms will be discussed in

Table 2.3: Papers using the hull model.

Abbr.	Paper Highlights	Fusion Level
ProNN [76]	Models each image set using the affine hull model with an assumption of Gaussian distribution. It maximizes the probability that each point belongs to its corresponding hull.	Decision
ISCRC [90]	Provides a collaborative representation framework for different hull models with different regularization. Includes the correlations between all gallery image sets.	Decision
CRNP [81]	Uses the same idea as ISCRC, but the reconstruction error is normalized by the nuclear norm of the image set and the l_2 norm of the combination coefficient.	Decision
SSDML [89]	Models each image set using restricted affine hull and extends the distance metric learning to the hull model.	Decision
SRN-ADML [56]	Adds the self regularized constraint on affine hull bases and the non-negativity constraint on sample coefficients. An adaptive distance metric learning strategy is also developed.	Decision
RNP [86]	Models each image set using a restricted affine hull model by simplifying the sparse regularization. A fast solver is also provided to speed up the computation.	Decision
SBDR [82]	Models each image set using a convex hull and proposes a set-based discriminative ranking model which optimizes the set-to-set distance and a projection feature space simultaneously.	Decision
SANS [34]	Models each image set using a restricted affine hull model and adds sparse constraints when calculating hull-based distance.	Decision

Section 2.2 to present how the discriminant analysis is conducted in the hull model. The set-based collaborative representation [90, 81] is closely related to the hull-based approaches. When a probe image set $\boldsymbol{P}$ is presented for classification, it tries to reconstruct it using the whole gallery,

$$\begin{aligned} &\min_{\boldsymbol{\alpha}_p, \boldsymbol{\alpha}_g} ||\boldsymbol{P}\boldsymbol{\alpha}_p - \boldsymbol{\mathcal{X}}\boldsymbol{\alpha}_g||_2^2 \\ &\text{s.t. } \sum_n \alpha_p^n = 1,\ \sum_n \alpha_g^n = 1,\ \alpha_p^n, \alpha_g^n \in \boldsymbol{\mathcal{R}}, \end{aligned} \tag{2.16}$$

where $\boldsymbol{\mathcal{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ...\boldsymbol{X}_{N_s}]$ is the concatenation of the whole gallery image sets, and $\boldsymbol{\alpha}_g = \left[\boldsymbol{\alpha}_g^1, \boldsymbol{\alpha}_g^2, ..., \boldsymbol{\alpha}_g^{N_s}\right]$ are the corresponding reconstruction coefficients for each gallery image set. It can employ any constraints $\boldsymbol{\mathcal{R}}$ for different hull models. The final distance between a probe sample $\boldsymbol{P}$ and a gallery image set $\boldsymbol{X}_i$ is defined as the corresponding reconstruction error,

$$\mathcal{D}\left(\boldsymbol{P}, Xi\right) = ||\boldsymbol{P}\boldsymbol{\alpha}_p - \boldsymbol{X}_i\boldsymbol{\alpha}_g^i||_2^2, \tag{2.17}$$

or some normalized reconstruction error based on (2.17).

2.1.4 Attention-based Model

Approaches in this category do not have model assumptions on an image set. Samples in the image set are treated independently, and different attentions are assigned to each sample. The results are integrated on the feature or decision level with different weights. Therefore, the set weighting or attention scheme is of paramount importance. Except for strategies like max or min fusion [50], majority voting [30], Chen et al. [9] proposed a dual regression based approach (DLRC).

In DLRC, the weights are learned to minimize a reconstruction error. Feng et al. [19] extended DLRC into pairwise linear-regression classification (PLRC). PLRC minimizes the reconstruction error of both related and unrelated image sets. Using the reconstruction based attention, the matching complexity is very high. Recent set-to-set matching methods [85, 47] employ deep-learning

Table 2.4: Papers using the attention-based model.

Abbr.	Paper Highlights	Fusion Level
MMDML [50]	Learns for each class a deep neural network to map image samples into non-linear space. The set-to-set distance is the smallest pair-wise sample-to-sample Euclidean distance.	Decision
TDRM [30]	Proposes to learn a deep reconstruction model for each class. The probe image set is tested sample by sample to the class of the smallest reconstruction error. The final decision is the majority vote for all samples in the probe.	Decision
MN [83]	Proposes use two attention module to learn the the "visual" quality and the "content" quality.	Feature
SFDL [49]	Proposes a method that can optimize the feature and dictionary simultaneously. Sample-wise testing results are fused using majority voting.	Decision
DLRC [9]	The attention between different images are obtained by minimum reconstruction error of the mean difference using the reverse concatenation of two feature matrix.	Decision
NAN [85]	Proposes to add attention module in deep net to fuse image set features.	Feature
PLRC [19]	Proposes to measure the similarity with both the minimum reconstruction error of the related class and the maximum reconstruction error of the unrelated classes.	Feature
QAN [47]	Proposes a network to measure the quality of each image and use the quality score to fuse image set features	Feature
DMK [63]	Proposes deep matching kernel for point-to-point local similarity and introduces anchor-based aggregation to integrate local similarity into a global one.	Decision

architectures to learn the set weighting scheme. Specifically, an image set is embedded into a single feature representation using the weighted average. Yang et al. [85] introduced the Neural Aggregation Network, in which the learned features are fed to an attention mechanism which organizes the input through accessing external memory. These features are then aggregated into a fixed length feature vector adaptively, through two attention blocks. Liu et al. [47] followed a different approach and proposed a quality-aware network (QAN) in which the template and image quality scores are learned jointly by minimizing a weighted triplet loss function. Xei et al. [83] proposed a multicolumn network (MN) which consists of two attention modules. The first attention module assesses the "visual" quality level according to the image itself, and the second module assesses the "content" qualities relative to the other images within the set. However, the training process of the above methods is quite complicated. First, template embedding and the attention scores are learned simultaneously. The learning of attention scores sometimes restricts and degrades the learning of deep feature representations. Second, set-based settings are introduced in every training batch. In particular, there should be several samples from the same class within a training batch to learn the attention score effectively. Third, some of the algorithms employed the pair-wised or triplet-based loss which is tricky to sample and difficult to converge.

2.1.5 Mixture Model

Approaches in this category use a mixture of multiple models to model an image set. This category can be further divided into two sub-categories: (i) mixture of same local models [73, 12, 10, 75]; (ii) mixture of different global models [51, 36, 65].

Mixture of Same Local Models: Approaches in this group model each image set using a mixture of the same type of local structures (e.g., mixture of linear subspace, mixture of Gaussian). The objective is to capture the variations in different local areas of the original image set. It mainly

Table 2.5: Papers using the local mixture model.

Abbr.	Paper Highlights	Fusion Level
HERML [36]	Models each image set uses a mixture of the mean, covariance matrix and Gaussian distribution. Embeds the Gaussian distribution and covariance matrix into higher Hilbert space and uses multi-kernel metric learning to fuse them.	Feature
KSL [65]	Converts existing distance metric from different image set modeling into kernel matrix and proposes a sparse kernel learning algorithm to automatically learn a sparse combination. For the subspace model, the MMD kernel is employed.	Feature
LMKML [51]	Models each image set using multi-order statistics and embeds them into Hilbert space using concatenation. Multi-kernel metric learning is proposed to fuse the distance calculated for different statistics.	Decision

faces three challenges: (i) how to partition data into local patches; (ii) how to compare between the local patches; and (iii) how to fuse the result from different local patches. Since all the local patches use the same type of model, the second challenge can be solved using any single model approach discussed in Section. 2.1.1, 2.1.2, and 2.1.3. It will not be discussed in this section. In the manifold-to-manifold distance algorithm, the maximum linear patch (MLP) [73] approach is proposed. It can search for the maximum local area that the linear constraint holds. The whole image set is then partitioned to a mixture of local linear subspaces. All the pairwise patch-to-patch distances are computed. A minimum rule is used to fuse these distances. The MLP is also used in [75] to initialize the local area for different Gaussian components. Distances between different components are fused using multi-kernel distance metric learning. Instead of applying MLP in every image set, MLP is only employed to partition the reference image set [12] into different local patches. Other image sets are aligned to the reference set patch by patch sequentially. As a result, the number of patch pairs is reduced from N^2 to N, where N is the number of patches

per set. In [10], a sparse representation-based approach is proposed to extract the m dimensional local subspaces. For each gallery image set, the single vector sparse representation is applied to each sample. The vectors corresponding to the m largest reconstruction coefficients are used to extract a local linear subspace. For a probe image set, subspaces are extracted from all possible combinations of m samples and only the one with the smallest distance to a gallery patch is kept. An average fusion rule is conducted across all the gallery patches within an image set.

In summary, algorithms in this group focus more on the local area partitioning. The fusion rules are usually simple minimum or average rules. Modeling each image set using a mixture of local models can capture the complex local structures of an image set. This model is more flexible than using a single global model. It is more suitable for large scale image sets with a complex data structure. However, this model is more complex and sensitive to the noise with high computational cost.

Mixture of Different Global Models: Approaches in this group use different models to model each image set globally. As a result, the only challenge is the fusion of different models. Multi-kernel distance metric (LMKML) learning is proposed in [51]. In this chapter, the localized multi-order statistical properties are used to model each image set. As the kernel functions are well-defined for each statistical property, the LMKML jointly learns a weight and a projection, while simultaneously integrating each component in the projected space. This LMKML is also used in hybrid Euclidean and Riemannian metric learning [36] to integrate the mean, covariance matrix, and Gaussian distribution. Sparse kernel learning (SKL) [12] is another approach to learn a sparse combination of the selected models.

In summary, using the mixture of different global models can integrate more information. Since different models have different advantages and limitations, they are mutually complementary. The mixture model can offer a complete view of the image set, but it is relatively sensitive to noise and

Table 2.6: Papers using the global mixture model.

Abbr.	Paper Highlights	Fusion Level
DARG [75]	Models each image set using GMM. Proposes multiple probabilistic kernels to embed Riemannian manifold of Gaussian into RKHS. Multi-kernel metric learning is employed to fuse the different components of GMM.	Decision
SANS [10]	Models each image set using a mixture of multiple linear subspaces. The local linear subspaces are extracted via sparse representation. The probe image set is assigned to the class with the lowest reconstruction error.	Feature
MMD [73]	Models each image set using a mixture of multiple linear subspaces defined on Maximal linear patches. The manifold-to-manifold distance is computed via integrating pair-wise subspace distances. Combines the Euclidean distance and projection metric into a new dissimilarity measurement.	Decision
SAVOR [12]	Models each image set using a mixture of multiple linear subspaces. To structure the subspaces, it selects a reference image set and aligns the local patches to it. The distances between corresponded subspaces are computed as similarity vectors.	Decision

computationally expensive.

2.2 Discriminant Analysis

In this section, the supervised measurements mentioned in Section 2.1 are summarized, and the associated discriminant analysis techniques are discuss. Although a variety of models are employed to provide unsupervised similarity measurements between image sets, it is not guaranteed that they

will fit properly for all the situation. To this end, discriminant analysis techniques are extended to these models to train a better similarity measurement for the classification task at hand.

A summary of different attributes for the approaches concerned is provided in Table 2.7. In most of the discriminant analysis techniques, these approaches are grouped according to the space where discriminant analysis is conducted (i.e. Euclidean space, Riemannian Manifold, and multi-modal discriminant analysis).

2.2.1 Euclidean Space

Approaches in this category employ models which lie in Euclidean space. As described in Section 2.1, the similarity measurements for hull model and attention-based approaches are in the Euclidean space. In the attention-based framework, any sample-based discriminant analysis technique can be applied directly without large changes. Therefore, only the discriminant analysis techniques employed in the hull-based model are discussed in this subsection. Specifically, distance metric learning [89, 56] and discriminative ranking [82] are covered.

The objective of distance metric learning [57] is to learn an accurate distance metric that reflects what is considered to be "similar" and "dissimilar" for a specific task. It has been extended to the hull-based distance model [89, 56] by introducing a semi-positive definite Mahalanobis matrix $\boldsymbol{M}$ into Equation (2.12),

$$\begin{aligned} &\mathcal{D}_{\boldsymbol{M}}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j\right)^T \boldsymbol{M}\left(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j\right) \\ &\text{s.t. } \sum_n \alpha_i^n = 1,\ \sum_n \alpha_j^n = 1,\ \alpha_i^n, \alpha_j^n \in \boldsymbol{\mathcal{R}}, \end{aligned} \tag{2.18}$$

where $\boldsymbol{\mathcal{R}}$ could be any hull-based restriction. If decomposing $\boldsymbol{M}$ into $\boldsymbol{L}^T\boldsymbol{L}$ via Cholesky decomposition, then the Mahalanobis metric becomes

$$\begin{aligned} \mathcal{D}_{\boldsymbol{M}}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) &= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} ||\boldsymbol{L}\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{L}\boldsymbol{X}_j\boldsymbol{\alpha}_j||_2^2 \\ \text{s.t. } \sum_n \alpha_i^n = 1,\ &\sum_n \alpha_j^n = 1,\ \alpha_i^n, \alpha_j^n \in \boldsymbol{\mathcal{R}}. \end{aligned} \tag{2.19}$$

It is equivalent to first project the all samples into a more discriminative space, and then calculate the hull-based distance. The Mahalanobis matrix $\boldsymbol{M}$ or $\boldsymbol{L}$ can be learned by minimizing an objective function

$$\min_{\boldsymbol{M}} \mathcal{J} = \mathcal{L}\left(\mathbb{G}, \boldsymbol{M}\right) + \mathcal{R}\left(\boldsymbol{M}\right), \tag{2.20}$$

where the first term $\mathcal{L}$ is a loss function defined to push dissimilar image sets farther apart and pull similar image sets closer. The second term, $\mathcal{R}$, is the regularization defined to avoid overfitting. Equation (2.20) is a joint optimization of the Mahalanobis matrix M and the combination coefficient $\boldsymbol{\alpha}$. Similar to the strategy adopted in multivariable optimization problems, they can be optimized alternatively. In different distance metric learning algorithms, different loss functions and regularization are defined. Theoretically, this framework can be applied to any hull-based distance model with any distance metric learning model. In set-to-set distance metric learning [89], the regularized affine hull model [86] is used. The Mahalanobis matrix is learned such that the distances between similar image sets are smaller than an upper boundary and the distances between dissimilar image sets are larger than a lower boundary. In [56], the self-regularization constrained affine hull model is used with adaptive large-margin distance metric learning.

The objective of discriminative ranking is to learn a desired ranking $\hat{\boldsymbol{r}}_{\boldsymbol{P}}$ of all the gallery image sets $\mathbb{G}$ with respect to a probe image set $\boldsymbol{P}$, such that any relevant gallery sets should be ranked before the irrelevant ones. A desired ranking model $\boldsymbol{M}$ is expected to distinguish the optimal ranking $\hat{\boldsymbol{r}}_{\boldsymbol{P}}$ from any other candidate ranking $\boldsymbol{r}_{\boldsymbol{P}}$. This ranking has been successfully employed

Table 2.7: Papers which employed discriminant analysis techniques in Euclidean space.

Abbr.	Gen.	Model	Constraints	Transform	Learning Strategy
DMK [63]	No	Attention	Global	Non-linear	Deep learning
TDRM [30]	No	Attention	Global	Non-linear	Deep learning
SFDL [49]	No	Attention	Local	Linear	Dictionary Learning
NAN [85]	Yes	Attention	Global	Non-linear	Deep learning
QAN [47]	Yes	Attention	Global	Non-linear	Deep learning
MN [83]	Yes	Attention	Global	Non-linear	Deep learning
DLRC [9]	No	Attention	Global	Linear	Dictionary Learning
PLRC [19]	No	Attention	Global	Linear	Dictionary Learning
RevTr [29]	Yes	Attention	Global	Linear	Binary Classification
SBDR [82]	Yes	Hull	Local	Non-linear	Metric Learning
SSDML [89]	Yes	Hull	Global	Linear	Metric Learning
MMDML [50]	No	Attention	Local	Non-linear	Metric Learning
SRN-ADML [56]	Yes	Hull	Local	Linear	Metric Learning

The "Model" column indicates the associated model for each image set, and possible entries are: (i) "Subspace", (ii) "Statistical", (iii) "Hull", and (iv) "Attention". The "Gen." column indicates whether it can be extended directly to unseen subjects/objects without retraining. The "Constraints" column records the type of constraints in the objective function of the respective methods. It is either "Global" or "Local".The "Transform" column records whether the learned transform is "Linear" or "Non-linear".

for sample-based recognition [55]. In addition, it is extended to set-based discriminative ranking [82], using the hull model. Generally, the optimal ranking can be inferred by

$$\hat{\boldsymbol{r}}_{\boldsymbol{P}} = arg\max\langle \boldsymbol{M}, \psi\left(\boldsymbol{P}, \mathbb{G}, \boldsymbol{r}_{\boldsymbol{P}}\right)\rangle, \tag{2.21}$$

where $\boldsymbol{M}$ is the learned ranking model, and $\psi\left(\boldsymbol{P}, \mathbb{G}, \boldsymbol{r}_{\boldsymbol{P}}\right)$ is the joint feature map defined on the probe image set $\boldsymbol{P}$, the whole gallery $\mathbb{G}$, and a candidate ranking $\boldsymbol{r}_{\boldsymbol{P}}$. There exist different designs for M and ψ. In the set-based discriminative ranking [82], it is transferred to the distance metric learning problem. The basic assumption is that a smaller distance corresponds to a higher ranking. In particular, the model $\boldsymbol{M}$ is a semi-positive definite Mahalanobis matrix, the feature mapping ψ

is defined as,

$$\psi\left(\boldsymbol{P}, \mathbb{G}, \boldsymbol{r}_{\boldsymbol{P}}\right)=\sum_{i \in S_{\boldsymbol{P}}^{+}} \sum_{j \in S_{\boldsymbol{P}}^{-}} s_{i, j}\left(\frac{\phi\left(\boldsymbol{P}, \boldsymbol{X}_{i}\right)-\phi\left(\boldsymbol{P}, \boldsymbol{X}_{j}\right)}{\left|S_{\boldsymbol{P}}^{+}\right| \cdot\left|S_{\boldsymbol{P}}^{-}\right|}\right), \tag{2.22}$$

where

$$s_{i j}=\begin{cases}1 & \boldsymbol{X}_{i} \text{ ranks before } \boldsymbol{X}_{j} \text{ in } \boldsymbol{r}_{\boldsymbol{P}} \\ -1 & \boldsymbol{X}_{i} \text{ ranks after } \boldsymbol{X}_{j} \text{ in } \boldsymbol{r}_{\boldsymbol{P}}\end{cases} \tag{2.23}$$

and

$$\phi\left(\boldsymbol{P}, \boldsymbol{X}_{i(j)}\right)=-\left(\boldsymbol{P}-\boldsymbol{X}_{i(j)}\right)\left(\boldsymbol{P}-\boldsymbol{X}_{i(j)}\right)^{T}. \tag{2.24}$$

Here, $S_{\boldsymbol{P}}^{+}$ and $S_{\boldsymbol{P}}^{-}$ denote the index set of relevant and irrelevant image sets, respectively. $\phi\left(\boldsymbol{P}, \boldsymbol{X}_{i(j)}\right)$ is a feature map characterizing the relationship between the probe set $\boldsymbol{P}$ and gallery set $\boldsymbol{X}_{i(j)}$. As a result,

$$\left\langle\boldsymbol{M}, \psi\left(\boldsymbol{P}, \mathbb{G}, \boldsymbol{r}_{\boldsymbol{P}}\right)\right\rangle=\sum_{i \in S_{\boldsymbol{P}}^{+}} \sum_{j \in S_{\boldsymbol{P}}^{-}} s_{i, j}\left(\frac{\mathcal{D}_{\boldsymbol{M}}\left(\boldsymbol{P}, \boldsymbol{X}_{i}\right)-\mathcal{D}_{\boldsymbol{M}}\left(\boldsymbol{P}, \boldsymbol{X}_{j}\right)}{\left|S_{\boldsymbol{P}}^{+}\right| \cdot\left|S_{\boldsymbol{P}}^{-}\right|}\right), \tag{2.25}$$

where $\mathcal{D}_{\boldsymbol{M}}\left(\boldsymbol{P}, \boldsymbol{X}_{i(j)}\right)$ is the Mahalanobis distance defined in Equation (2.18). The model $\boldsymbol{M}$ can be obtained using any distance metric learning strategy.

2.2.2 Riemannian Manifold

Approaches in this category employ models which lie in the Grassman Manifold or the SPD Manifold. For both, the Riemannian geometry holds. The Riemannian geometry is well-explored with different discriminant analysis techniques, including kernel learning [66], dictionary learning [24], graph embedding framework [28, 71] and distance metric learning [38, 25, 39].

As discussed in Section 2.1.1 and Section 2.1.2, the similarity measurement under the Riemannian geometry is conducted via mapping features into a RKHS with a corresponding kernel matrix

Table 2.8: Papers which employed discriminant analysis techniques in Riemannian space.

Abbr.	Gen.	Model	Constraints	Transform	Learning Strategy
SPDNet[35]	Yes	Statistical	Global	Non-linear	Deep Learning
PML [38]	Yes	Subspace	Global	Non-linear	Metric Learning
kSLCC [24]	No	Statistical	Local	Non-linear	Dictionary learning
LEML [39]	Yes	Statistical	Global	Non-linear	Metric learning
$\mathcal{K}_{log}^{ploy}$ [66]	No	Subspace	Global	Non-linear	Kernel Learning
GDL [26]	No	Subspace	Global	Non-linear	Dictionary learning
GGDA [28]	Yes	Subspace	Local	Non-linear	Graph Embedding
SPD-ML [25]	Yes	Statistical	Local	Non-linear	Metric Learning
CDL [71]	Yes	Statistical	Global	Non-linear	Graph Embedding
DCLR [77]	Yes	Statistical	Global	Non-linear	Deep Learning

The "Model" column indicates the associated model for each image set, and possible entries are: (i) "Subspace", (ii) "Statistical", (iii) "Hull", and (iv) "Attention". The "Gen." column indicates whether it can be extended directly to unseen subjects/objects without retraining. The "Constraints" column records the type of constraints in the objective function of the respective methods. It is either "Global" or "Local". The "Transform" column records whether the learned transform is "Linear" or "Non-linear".

$\mathcal{K}$. Instead of a defined kernel, the joint kernel and classifier learning framework [66] proposes to learn simultaneously the kernel matrix $\mathcal{K}$ and classification model $\mathcal{W}$ via minimizing a loss function

$$\min_{\mathcal{W},\mathcal{K}} \mathcal{J} = \mathcal{L}\left(\mathcal{W}, \mathcal{K}\right) + \lambda \mathcal{R}\left(\mathcal{K}\right), \tag{2.26}$$

where the first term could be the loss function of any classifier characterized by $\mathcal{W}$. It is designed for better classification performance. The second term is designed to preserve the Riemannian structure. A general principle is that the distance in the mapped space should be as close as possible to the distance in the original manifold. It can be considered as regularization in Equation (2.20).

In Riemannian coding and dictionary learning [24], a general coding framework under the Riemannian geometry is proposed. Let ϕ be a mapping to an RKHS induced by the kernel

$\boldsymbol{\mathcal{K}}\left(\boldsymbol{X}_i, \boldsymbol{X}_j\right) = \phi(\boldsymbol{X}_i)^T \phi(\boldsymbol{X}_j)$. The general coding framework is

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \left\| \phi\left(\boldsymbol{P}\right) - \sum_{j=1}^{N} \alpha_j \phi\left(\boldsymbol{D}_j\right) \right\|_2^2 + \lambda \gamma\left(\boldsymbol{\alpha}; \phi\left(\boldsymbol{P}\right), \phi\left(\boldsymbol{D}\right)\right) \\ &= \min_{\boldsymbol{\alpha}} \left[\boldsymbol{\mathcal{K}}\left(\boldsymbol{P}, \boldsymbol{P}\right) - 2\boldsymbol{\alpha}^T \boldsymbol{\mathcal{K}}\left(\boldsymbol{P}, \boldsymbol{D}\right) + \boldsymbol{\alpha}^T \boldsymbol{\mathcal{K}}\left(\boldsymbol{D}, \boldsymbol{D}\right) \boldsymbol{\alpha} \right] + \lambda \gamma\left(\boldsymbol{\alpha}; \boldsymbol{\mathcal{K}}\right) \\ &\text{s.t.} \quad \boldsymbol{\alpha} \in \boldsymbol{\mathcal{R}}, \end{aligned} \tag{2.27}$$

where the first term is designed to minimize the reconstruction error, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N] \in \mathbb{R}^N$ is the vector of codes and $\boldsymbol{D}$ denotes the dictionary on manifold. The second term γ is a prior on the codes which can be considered as the regularization in Equation (2.20), and $\boldsymbol{\mathcal{R}}$ is a set of constraints on α. It is a general framework that can be applied to the subspace model and the SPD model with the corresponding kernels $\boldsymbol{\mathcal{K}}$. Some typical examples of the general framework include, (i) Kernel Sparse Coding: It can be obtained from Equation (2.27) by defining

$$\gamma\left(\boldsymbol{\alpha}; \boldsymbol{P}, \boldsymbol{D}\right) = \|\boldsymbol{\alpha}\|_1, \ \boldsymbol{\mathcal{R}} = U \tag{2.28}$$

without any constraint; and (ii) Locality-Constrained Linear Coding: It can be obtained with

$$\gamma\left(\boldsymbol{\alpha}; \boldsymbol{P}, \boldsymbol{D}\right) = \sum_j \left(exp\left(\sigma \|\boldsymbol{P} - \boldsymbol{D}_j\|\right) \alpha_j\right)^2, \ \boldsymbol{\mathcal{R}} = \left\{ \boldsymbol{\alpha} | \sum_j \alpha_j = 1 \right\}. \tag{2.29}$$

Beside reconstructing the training samples accurately, the generated codes are also expected to be discriminative for the classification task. To this end, the supervised training data can be efficiently employed in the framework discussed above. The codes $\boldsymbol{\alpha}_i$, dictionary $\boldsymbol{D}$, and a classifier $\mathcal{W}$ can be optimized jointly via

$$\min_{\mathcal{W}, \boldsymbol{D}, \boldsymbol{\alpha}_i} \sum_{i=1}^{N_s} \mathcal{L}_\phi\left(\boldsymbol{D}, \boldsymbol{\alpha}_i, \boldsymbol{X}_i\right) + \lambda \sum_{i=1}^{N_s} \mathcal{L}_{\mathcal{W}}\left(\mathcal{W}, l_i, \boldsymbol{\alpha}_i\right), \tag{2.30}$$

where $\mathcal{L}_\phi$ denotes the loss defined in Equation (2.27), the first term is the loss across the whole training set. $\mathcal{L}_{\mathcal{W}}$ could be the loss function of any classifier. The classification is based on the codes $\boldsymbol{\alpha}_i$ corresponding to $\boldsymbol{X}_i$.

Although the graph embedding discriminant analysis (GGDA) [28] and covariance discriminant learning (CDL) [71] are designed for Grassman manifold and SPD manifold, respectively, they can be unified to a kernel based graph embedding discriminant analysis framework. The graph embedding framework seeks to map the original data $\boldsymbol{X}$ into a more discriminant space $\phi(\boldsymbol{X})$ and preserve the similarity structure between pairs of image sets. A graph $(\boldsymbol{X}, \boldsymbol{W})$ is used to capture this similarity structure, where $\boldsymbol{W} \in \mathbb{R}^{N_s \times N_s}$ is a symmetric matrix with $\boldsymbol{W}(i,j)$ is the similarity between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$. In GGDA, the local geometry structure is captured using within class similarity graph $\boldsymbol{W}_w$ and between class similarity graph $\boldsymbol{X}_b$. The simplest way is

$$\begin{aligned} \boldsymbol{W}_w(i,j) &= \begin{cases} 1 & \boldsymbol{X}_i \in N_w(\boldsymbol{X}_j) \text{ or } \boldsymbol{X}_j \in N_w(\boldsymbol{X}_i) \\ 0 & \text{otherwise} \end{cases} \\ \boldsymbol{W}_b(i,j) &= \begin{cases} 1 & \boldsymbol{X}_i \in N_b(\boldsymbol{X}_j) \text{ or } \boldsymbol{X}_j \in N_b(\boldsymbol{X}_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.31}$$

where $N_w(\boldsymbol{X}_i)$ is the local neighbor set that shares the same label with $\boldsymbol{X}_i$ and $N_b(\boldsymbol{X}_i)$ contains neighbors that have different labels with $\boldsymbol{X}_i$. In CDL, only the within class similarity graph is included and is defined as

$$\boldsymbol{W}_w(i,j) = \begin{cases} 1/m_k & l_i = l_j = k \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

where m_k is the number of samples in the kth class. The difference is that in Equation (2.31) the local neighborhood structure is preserved. It can employ the objective function of any distance metric learning algorithms with kernel trick. Confining the solution to be linear, it becomes

$$\begin{aligned} \phi(\boldsymbol{X}_i) &= \left[\langle \boldsymbol{\alpha}_1, \boldsymbol{X}_i \rangle, \langle \boldsymbol{\alpha}_2, \boldsymbol{X}_i \rangle, ..., \langle \boldsymbol{\alpha}_r, \boldsymbol{X}_i \rangle\right]^T \\ \boldsymbol{\alpha}_r &= \sum_{j=1}^{N_s} a_r^j \boldsymbol{X}_j \end{aligned} \tag{2.33}$$

where r is the dimension of the projected space. Defining

$$\boldsymbol{A} = \begin{pmatrix} a_1^1 & a_1^2 & \cdots & a_1^{N_s} \\ a_2^1 & a_2^2 & \cdots & a_2^{N_s} \\ \vdots & \vdots & \ddots & \vdots \\ a_r^1 & a_r^2 & \cdots & a_r^{N_s} \end{pmatrix} \tag{2.34}$$

and

$$\boldsymbol{k}_i = [\boldsymbol{\mathcal{K}}(\boldsymbol{X}_i, \boldsymbol{X}_1), \boldsymbol{\mathcal{K}}(\boldsymbol{X}_i, \boldsymbol{X}_2), ..., \boldsymbol{\mathcal{K}}(\boldsymbol{X}_i, \boldsymbol{X}_{N_s})]^T, \tag{2.35}$$

it becomes

$$\phi(\boldsymbol{X}_i) = \boldsymbol{A}\boldsymbol{k}_i. \tag{2.36}$$

It can employ any objective function that involves the similarity matrix to optimize the coefficients $\boldsymbol{A}$. In GGDA, $\boldsymbol{A}$ is optimized via maximizing the distance between target similar image sets and minimizing the distance between dissimilar image sets. In CDL, the objective functions of LDA and PLS are employed. Because both algorithms are kernel based, any kernel described in Section 2.1.1 and Section 2.1.2 can be employed theoretically.

All the algorithms discussed above first project the data onto a reduced Hilbert space, then conduct discriminant analysis. In contrast, the projection metric learning (PML) [38], log-Euclidean metric learning (LEML) [39], and semi-positive definite metric learning (SPD-ML) [25] seek to learn a generic mapping directly from manifold to a lower order and more discriminative manifold. The mapping $\phi(\boldsymbol{L}, X_i)$ is parameterized by a full rank matrix $\boldsymbol{L} \in \mathbb{R}^{d \times r}$. The distance in the projected space becomes $\mathcal{D}\left(\phi(\boldsymbol{L}, X_i), \phi(\boldsymbol{L}, X_j)\right)$, where the distance function $\mathcal{D}$ could be calculated using any kernel under the Riemannian geometry. To learn a discriminative mapping, the general objective is to push the similar image sets closer and pull the dissimilar image sets farther apart. The objective functions can be designed for the problem at hand.

In PML, it is proposed to map the linear subspace $\boldsymbol{Y}_i \in \boldsymbol{\mathcal{G}}(m,d)$ ($\boldsymbol{Y}_i$ is the subspace representation of $\boldsymbol{X}_i$) onto a lower order Grassman manifold $\boldsymbol{L}^T\boldsymbol{Y}_i \in \boldsymbol{\mathcal{G}}(m,r)$. To ensure that the projected subspace is on the Grassman manifold, the orthogonal component of $\boldsymbol{L}^T\boldsymbol{Y}_i$ defined by $\boldsymbol{L}^T\boldsymbol{Y}'_i$ is used. The distance between two subspaces $(\boldsymbol{Y}_i, \boldsymbol{Y}_j)$ in the projected space is defined as:

$$\begin{aligned}\mathcal{D}^2\left(\phi(\boldsymbol{L},\boldsymbol{X}_i),\phi(\boldsymbol{L},\boldsymbol{X}_j)\right) &= 2^{1/2}\|\boldsymbol{L}^T\boldsymbol{Y}'_i\boldsymbol{Y}_i'^T\boldsymbol{L} - \boldsymbol{L}^T\boldsymbol{Y}'_j\boldsymbol{Y}_j'^T\boldsymbol{L}\|_F^2 \\ &= tr\left(\boldsymbol{Q}\boldsymbol{A}_{ij}\boldsymbol{A}_{ij}^T\boldsymbol{Q}\right),\end{aligned} \tag{2.37}$$

where $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^T$ and $\boldsymbol{A}_{ij} = \boldsymbol{Y}'_i\boldsymbol{Y}_i'^T - \boldsymbol{Y}'_j\boldsymbol{Y}_j'^T$. The objective function is designed to maximize the average between class distance, and meanwhile minimize the average within class distance. The projection matrix $\boldsymbol{Q}$ and the orthogonal bases $\boldsymbol{Y}'$ are updated alternatively.

Similarly, SPD-ML proposes to learn a mapping from the original SPD manifold to a lower order SPD manifold. In particular, the mapping is defined as

$$\phi(\boldsymbol{L},\boldsymbol{X}_i) = \boldsymbol{L}^T\boldsymbol{X}_i\boldsymbol{L}\,. \tag{2.38}$$

SPD-ML uses the AID (Equation (2.5)) and Stein metric (Equation (2.6)) in the projected space. To learn a discriminative projection, SPD-ML employs the graph embedded framework to maximize the distances between target imposters (image sets under different labels) and minimize the distances between target neighbors (image sets sharing the same labels).

LEML also seeks to learn a mapping from the original SPD manifold to a lower order SPD manifold, but it is based on the log-Euclidean distance. The mapping is defined as:

$$\phi(\boldsymbol{L},\boldsymbol{X}_i) = \boldsymbol{L}^T log(\boldsymbol{X}_i)\boldsymbol{L}\,, \tag{2.39}$$

where $log()$ is the ordinary matrix logarithm operation. The log-Euclidean distance in the projected space then becomes

$$\begin{aligned}\mathcal{D}_L\left(\phi(\boldsymbol{L},\boldsymbol{X}_i),\phi(\boldsymbol{L},\boldsymbol{X}_j)\right) &= \|\boldsymbol{L}^T log(\boldsymbol{X}_i)\boldsymbol{L} - \boldsymbol{L}^T log(\boldsymbol{X}_j)\boldsymbol{L}\|_F^2 \\ &= tr\left(\boldsymbol{Q}\left(log(\boldsymbol{X}_i) - log(\boldsymbol{X}_j)\right)\left(log(\boldsymbol{X}_i) - log(\boldsymbol{X}_j)\right)\right)\end{aligned} \tag{2.40}$$

where $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{L}\boldsymbol{L}^T$ is a rank-r PSD matrix of size $d \times d$. The distance described in Equation (2.40) is in the form of a Mahalanobis-like distance. LEML then employ the objective function of information-theoretic metric learning [13] to learn the SPD matrix Q.

In summary, the discriminant analysis under the Riemannian geometry utilizes either the manifold-to-manifold embedding or the manifold-to-Hilbert embedding.

2.2.3 Multi-Modal Discriminant Analysis

Approaches in this category conduct discriminant analysis across multiple modalities X_i^t, where the superscript t indexes different modalities. The multi-modal here indicates same models in different local areas or different global models, which correspond to the two types of mixture models discussed in Section 2.1.5. In general, it seeks to learn a discriminative mapping $\phi(*)$ and sometimes a weight w^t to fuse the measurements together.

In particular, DARG [75] employs the mixture Gaussian model and represents each image set $\boldsymbol{X}_i$ as $\left\{g_i^t, w_i^t\right\}_{t=1}^{N_i}$, where g_i^t denotes the t-th Gaussian component, w_i^t is the weight, and N_i is the number of Gaussian components for $\boldsymbol{X}_i$. The weights are estimated when extracting the GMM model. DARG extends kernel discriminant analysis into a weighted version which can be formulated using the kernel trick Equation (2.33) and Equation (2.34) by

$$\max_{\boldsymbol{A}} \frac{|\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A}|}{|\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}|}, \tag{2.41}$$

where the between class inertia $\boldsymbol{B}$ and within class inertia $\boldsymbol{W}$ are defined as

$$\begin{aligned} \boldsymbol{B} &= \sum_{i=1}^{N_c} N_i \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^T, \\ \boldsymbol{W} &= \sum_{i=1}^{N_c} \frac{1}{w_i} \sum_{t=1}^{N_i} \left(\boldsymbol{k}_i^t - \boldsymbol{\mu}_i\right)\left(\boldsymbol{k}_i^t - \boldsymbol{\mu}_i\right)^T, \end{aligned} \tag{2.42}$$

and

$$\begin{aligned}
&\boldsymbol{\mu}_i = \frac{1}{N_i w_i}\sum_{t=1}^{N_i} w_i^t k_i^t, \quad \boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N_c}\frac{1}{w_i}\sum_{t=1}^{N_i} w_i^t k_i^t \\
&w_i = \sum_{t=1}^{N_i} w_i^t, \quad k_i^t = [\boldsymbol{\mathcal{K}}(g_i^t, g_1), \boldsymbol{\mathcal{K}}(g_i^t, g_2), ..., \boldsymbol{\mathcal{K}}(g_i^t, g_{N_s})]^T.
\end{aligned} \tag{2.43}$$

LMKML [51] represents each image set using a multi-order statistical model $\left\{\mathbf{x}_i^t\right\}_{t=1}^N$, where $\mathbf{x}_i^t$ denotes the feature vector from the t-th order statistics, and N is the number of statistics included. It extends distance metric learning to kernel discriminant analysis, where the weighted distance in the projected space is defined as

$$\begin{aligned}
&\mathcal{D}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) \\
&= \sum_{t=1}^{N} w^t\left(\phi(\mathbf{x}_i^t)\right)\left(\phi(\mathbf{x}_i^t) - \phi(\mathbf{x}_j^t)\right)^T \boldsymbol{M}\left(\phi(\mathbf{x}_i^t)\right)\left(\phi(\mathbf{x}_i^t) - \phi(\mathbf{x}_j^t)\right) w^t\left(\phi(\mathbf{x}_j^t)\right),
\end{aligned} \tag{2.44}$$

where the Mahalanobis matrix $\boldsymbol{M}$ can be decomposed into $\boldsymbol{W}\boldsymbol{W}^T$, and the weighting function $w^t\left(\phi(\mathbf{x}_i^t)\right)$ is designed as:

$$w^t\left(\phi(\mathbf{x}_i^t)\right) = \frac{exp(\boldsymbol{h}_t^T\mathbf{x}_i^t + b_t)}{\sum_{t=1}^N exp(\boldsymbol{h}_t^T\mathbf{x}_i^t + b_t)}. \tag{2.45}$$

The weighting function $\boldsymbol{h}_t$, b_t and Mahalanobis matrix $\boldsymbol{M}$ are optimized via

$$\max_{\boldsymbol{h}_t, b_t, \boldsymbol{M}} \sum_{(i,j)\in\boldsymbol{\mathcal{S}}_i} \frac{\mathcal{D}(\phi(\mathbf{x}_i^t), \phi(\mathbf{x}_j^t))}{|\boldsymbol{\mathcal{S}}_i|} - \sum_{(i,j)\in\boldsymbol{\mathcal{V}}_i} \frac{\mathcal{D}(\phi(\mathbf{x}_i^t), \phi(\mathbf{x}_j^t))}{|\boldsymbol{\mathcal{V}}_i|}, \tag{2.46}$$

where the first term is the average distance between similar image sets, and the second term is the average distance between dissimilar image sets. $\boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{\mathcal{V}}_i$ are the index sets for similar and dissimilar image sets, respectively, using the same kernel trick as in Eq (2.33).

Similar to LMKML, HERML [36] represents each image set using a hybrid statistical model $\left\{\boldsymbol{X}_i^t\right\}_{t=1}^N$, where $\boldsymbol{X}_i^t$ denotes the representation from the t-th statistic, and N is the number of modalities used. Unlike LMKML, which represents each statistic in a Euclidean space, HERML

represents image sets in a hybrid Euclidean and Riemannian space with corresponding kernels. The first order statistic lies in the Euclidean space. The second order statistic lies on an SPD manifold. The Gaussian components are embeded to a higher order SPD manifold. Instead of using the objective functions in Equation (2.46), it generates the ITML into multi-kernel metric learning using the same kernel trick as Equation (2.33).

Table 2.9: Papers which employed discriminant analysis techniques across multiple modalities.

Abbr.	Gen.	Model	Constraints	Transform	Learning Strategy
DARG [75]	Yes	Mixture	Local	Non-linear	Kernel Learning
HERML [36]	Yes	Mixture	Global	Non-linear	Metric Learning
KSL [65]	Yes	Mixture	Global	Non-linear	Kernel Learning
LMKML [51]	Yes	Mixture	Global	Non-linear	Metric Learning

The "Model" column indicates the associated model for each image set. The "Gen." column indicates whether it can be extended directly to unseen subjects/objects without retraining. The "Constraints" column records the type of constraints in the objective function of the respective methods. It is either "Global" or "Local". The "Transform" column records whether the learned transform is "Linear" or "Non-linear".

In summary, the multi-modal discriminant analysis mainly relies on the distance metric learning framework with kernels from different modalities.

2.3 Performance Prediction

In the field of performance prediction, researchers analyze the performance of a specific face recognition system under different circumstances. The performance prediction can be applied in: (i) multi-model fusion, (ii) quality control in enrollment, and (iii) sample ranking and selection. A performance prediction system contains mainly three parts: (i) the input, (ii) the output (i.e. different performance measurements), and (iii) the models that capture the mapping from input to

output. Regarding the input, existing literature extracts input features from (i) image quality features, or (ii) similarity score distributions. In the first group, external image quality assessors (IQA) are used to estimate image quality scores. Aggarwal et al. [2] propose to use image-specific (e.g., sharpness and saturation) and face-specific (e.g., poses and expressions) features. Beveridge et al. [6, 5, 4] useed a large number of subject-related covariates (e.g., ages, races, and genders) and image-related covariates (e.g., focus and resolution) as input features. Dutta et al. [17] focused only on poses, noise and blurry and propose a generative model to model the recognition performance distribution in a small image quality space. Deshpande et al. [15] extracted focus measure, brightness, obscured face and studied their influence on the accuracy of face recognition. Then a deep neural network was trained to build a binary classifier to reject low-quality images for recognition. There were mainly two limitations for the image quality based inputs. First, there are a lot of quality-related factors can be employed as input features and they are not independent with each other. Second, to get access to the quality features, external IQAs were employed which also introduced errors. Algorithms in the second group assumed that the overlapping regions between genuine and imposter score distributions reflect the recognition performance. Wang et al. [70] used the similarity score distributions to model the intrinsic and extrinsic factors of the performance of a face recognition system, while Klare et al. [42] proposed features derived from imposter similarity score distribution. One of the major limitations of estimating the recognition performance from similarity distribution is that it requires access to a large amount of data in new circumstances and cannot reflect the performance variations of each single image.

2.4 Binary Face Embedding

The task of binary face embedding is to represent single or multiple face images using a binary template. Existing binary encoding algorithms can be grouped into two categories, according to the

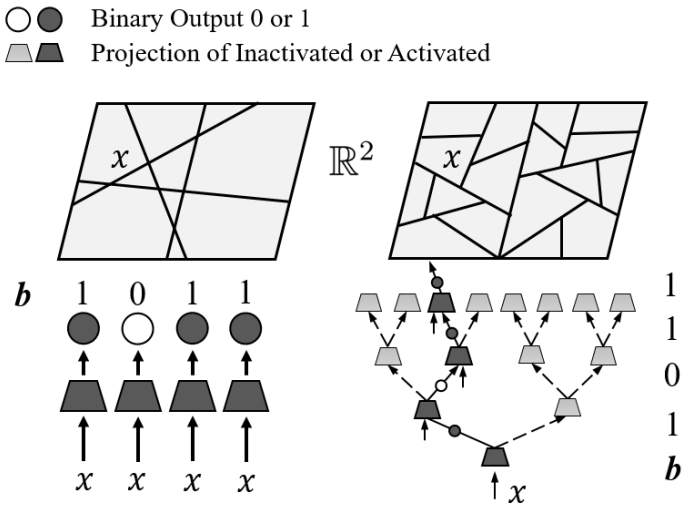

Figure 2.6: Comparison of the sequential code and tree-structure code. Without loss of generality, features in a two-dimensional real value feature space $\mathbb{R}^2$ (the grey parallelogram) are embedded into a four-bit binary code $[b_1, b_2, b_3, b_4]$, via multiple projections. The solid lines on the rhombus denote for the projections, which divide the original feature space into different regions. Points lie in the same region will share the same binary code. (L): In the sequence code, each bit of the output is learned from an independent projection. Using four projections, the real value feature space can be divided into at most 11 classes. (R): In the tree-structured code, each bit is learned recursively. There are 15 projections, and the real feature space is divided into 16 classes.

code structure: (i) the sequential code [16, 18], and (ii) the tree-structured code [67], as illustrated in Figure 2.6. The sequential code is learned via a sequence of projections bit by bit. The projections for one bit are independent of others. The tree-structured code is learned recursively through a binary tree, and each node on the tree is a projection. The projection of current bit depends on the output of the projection on its parent node. The tree-structured code achieves increased recognition power by dividing the original feature space into more classes. However, the number of projections to learn is increasing exponentially with the code length.

To learn a binary template, a straightforward approach is to learn a real-valued feature representation first. Then, the binary hashing is applied via different learning and quantization techniques. A review of the classic binary hashing techniques has been offered by Grauman et al. [20]. With the recent development and achievement of deep learning techniques, the hashing layer is introduced to learn the binary code in an end-to-end manner. Fan et al. [18] proposed adding rounding errors as regularization to jointly minimize the recognition loss and the rounding loss. Using the fully connected layers, the projection for each bit of the output code is independent of each other. Similar to other sequential codes, the recognition power is restricted. Instead of using a one-step optimization, Yury et al. [67] proposed a two-step approach via CNN and boosted hashing forest. The CNN learns discriminative real-valued feature vectors, while the boosted hashing forest learns binary code bit by bit recursively using multiple tree-structured projections. The identification and verification-based loss functions are designed to enhance the recognition performance. However, the number of projections to learn is increasing exponentially with the code length, and the random forest-based structure makes it difficult to optimize jointly with the CNN.

Most of the face hashing algorithms designed for image set follow an integrate-and-hash approach [16, 45, 59]. The image set is integrated into a single real-valued template on the feature level, and then the binary hashing is applied on the integrated template. Finally, the Hamming distance is used for the dissimilarity measurement. In addition to the limitations mentioned for the sequential code, template aggregation in early stages also results in the loss of discriminative information. Moreover, the binary templates for face image sets are evaluated only in easy datasets with easy protocols. In the existing literature, the evaluation is conducted in datasets [16, 45, 59] with a limited number of subjects, and subjects overlapping between the training and testing phases (e.g., The Big Bang Theory [45], and Prison Break [16]).

Chapter 3

Objective 1: Compact Templates and Robust Similarity Measurements

In this chapter, the objective is to develop an algorithm that generates compact templates and robust similarity measurements for set-to-set matching. In this work, the hull model is employed, because it better represents the within set variations. As described in Chapter 2, Section 2.1.3, previous approaches employ all the images in a set to span a hull. As a result, it contains redundant information and is sensitive to outliers. To address these limitations, the set-based prototype and metric learning framework (SPML) is proposed to (i) represent each image set with fewer but more discriminative templates, and (ii) learn a more accurate distance measurement for set-to-set matching. The objective of the prototype learning component of the framework is to represent the gallery-image set by using fewer templates while maintaining or improving the recognition performance. Each gallery-image set is then modeled as a hull spanned by the prototypes learned. To accurately reflect the notion of similarity when matching a probe with the learned prototypes, a Mahalanobis distance metric is jointly learned. To this end, the optimization problem is formulated

using a single loss function that jointly learns the prototypes and metric learning. Specifically, it brings similar image sets closer to each other, while pushing dissimilar ones far away, as illustrated by Figure 3.1. The primary contribution is a method with the following advantages: (i) it uses fewer prototypes to represent each gallery image set, reducing the computational cost and storage requirement; (ii) it increases the robustness of the hull model; and (iii) it can be used in conjunction with any hull model and any distance metric learning objective function.

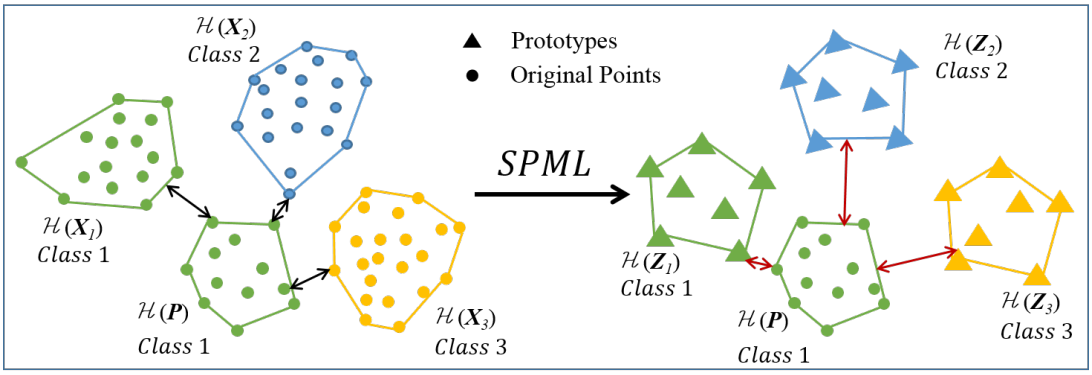

Figure 3.1: Illustration of set-based prototype and metric learning. (L): The $\mathcal{H}(\boldsymbol{X}_1)$, $\mathcal{H}(\boldsymbol{X}_2)$, and $\mathcal{H}(\boldsymbol{X}_3)$ denote three gallery sets from three different classes, while $\mathcal{H}(\boldsymbol{P})$ denotes a probe set. The $\mathcal{H}(\boldsymbol{X}_1)$ and $\mathcal{H}(\boldsymbol{P})$ belong to the same class. (R): The $\mathcal{H}(\boldsymbol{Z}_1)$, $\mathcal{H}(\boldsymbol{Z}_2)$, and $\mathcal{H}(\boldsymbol{Z}_3)$ denote the prototypes learned for the corresponding gallery sets. As illustrated, there are fewer samples in prototype presentation. After the process of SPML, distances between similar sets are "smaller", while the distances between dissimilar sets are "larger".

3.1 Method

The proposed SPML is developed based on the regularized affine hall (RAH) [86] model. In particular, an image set, $\boldsymbol{X}_i$, can be represented as a regularized affine hull (RAH), spanned by all

its samples:

$$\mathcal{H}(\boldsymbol{X}) = \left\{ \boldsymbol{X}\boldsymbol{\alpha} \middle| \sum_{n=1}^{N} \alpha_n = 1, \|\boldsymbol{\alpha}\|_{l_p} < \sigma \right\}, \tag{3.1}$$

with a regularization on the l_p norm of the the combination coefficient $\|\boldsymbol{\alpha}\|_{l_p} < \sigma$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N]^T$. The distance between two image sets $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is then defined as the geodesic distance between $\mathcal{H}(\boldsymbol{X}_i)$ and $\mathcal{H}(\boldsymbol{X}_j)$,

$$\begin{aligned} &\mathcal{D}^2(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left[(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j)^T (\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j) \right] \\ &s.t.\ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1,\ \|\boldsymbol{\alpha}_j\|_{l_p} < \sigma_2, \sum_{n=1}^{N_i} \alpha_n^i = 1, \sum_{n=1}^{N_j} \alpha_n^j = 1. \end{aligned} \tag{3.2}$$

By relaxing $\sum_{n=1}^{N_i} \alpha_n^i = 1$ and $\sum_{n=1}^{N_j} \alpha_n^j = 1$ to $\sum_{n=1}^{N_i} \alpha_n^i \approx 1$ and $\sum_{n=1}^{N_j} \alpha_n^j \approx 1$ and using the Lagrangian formulation, Equation (6.1.2) with $l_p = 2$ can be integrated as

$$\begin{aligned} &\mathcal{D}^2(\boldsymbol{X}_i, \boldsymbol{X}_j) \\ &= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left(\|\mathbf{u} - \hat{\boldsymbol{X}}_i\boldsymbol{\alpha}_i - \hat{\boldsymbol{X}}_j\boldsymbol{\alpha}_j\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}_i\|_2^2 + \|\boldsymbol{\alpha}_j\|_2^2 \right), \end{aligned} \tag{3.3}$$

where $\mathbf{u} = [\mathbf{0}; \mathbf{1}; \mathbf{1}]$, $\hat{\boldsymbol{X}}_i = \left[\boldsymbol{X}_i; \mathbf{1}^T; \mathbf{0}^T\right]$, $\hat{\boldsymbol{X}}_j = \left[-\boldsymbol{X}_j; \mathbf{0}^T; \mathbf{1}^T\right]$, and the column vectors $\mathbf{0}$ and $\mathbf{1}$ have the appropriate sizes associated with their corresponding context.

In the proposed prototype representation, each gallery image set, $\boldsymbol{X}_i$, is then represented as an regularized affine hall [86] spanned by the prototypes:

$$\mathcal{H}(\boldsymbol{Z}_i) = \left\{ \boldsymbol{Z}_i\boldsymbol{\beta} \middle| \sum_{k=1}^{K} \beta_k = 1, \|\boldsymbol{\beta}\|_{l_p} < \sigma \right\}, \tag{3.4}$$

where $\boldsymbol{Z}_i = \left[\boldsymbol{z}_1^i, \boldsymbol{z}_2^i, ..., \boldsymbol{z}_K^i\right] \in \mathbb{R}^{d \times K}$ denotes for the prototype set containing K prototypes ($K < N_i$). To distinguish between the representation coefficients of the original RAH $\mathcal{H}(\boldsymbol{X}_i)$, $\boldsymbol{\beta}_i = [\beta_1^i, \beta_2^i, ..., \beta_K^i]^T$ is used to denote the representation coefficients of the prototype RAH $\mathcal{H}(\boldsymbol{Z}_i)$.

Given a gallery $\mathbb{G} = \{(\boldsymbol{X}_i, l_i)|l_i \in [1, N_C]\}$, where l_i is the class label of $\boldsymbol{X}_i$, the prototypes, $\mathbb{Z}$, and the Mahalanobis distance metric, $\boldsymbol{M}$, are optimized by minimizing a loss function across the whole gallery,

$$\begin{aligned}(\mathbb{Z}, \boldsymbol{M}) &= arg\min_{\mathbb{Z},\boldsymbol{M}} \mathcal{L}(\mathbb{G}, \mathbb{Z}, \boldsymbol{M}) \\ &= arg\min_{\mathbb{Z},\boldsymbol{M}} \sum_{\mathbb{G}} \mathcal{L}_i(\boldsymbol{X}_i, \mathbb{Z}, \boldsymbol{M}),\end{aligned} \tag{3.5}$$

where $\mathbb{Z}$ denotes the prototype gallery contains all the corresponding prototype image set $\boldsymbol{Z}_i$, and $\boldsymbol{M}$ is a semi-positive-definite matrix. The proposed loss function $\mathcal{L}$ is a variant of the Large Margin Nearest Neighbors (LMNN) approach [78], and the loss on each gallery set $\boldsymbol{X}_i$ is defined as:

$$\begin{aligned}\mathcal{L}_i(\boldsymbol{X}_i, \mathbb{Z}, \boldsymbol{M}) =&(1-\mu)\sum_{\boldsymbol{\mathcal{S}}_i} \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) \\ &+\mu\sum_{\boldsymbol{\mathcal{V}}_i}[2\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) - \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_r)]_+.\end{aligned} \tag{3.6}$$

The objective of the first term is to pull target neighbors (i.e. $\boldsymbol{Z}_j$) "closer", where target neighbors denote the k-nearest prototype sets to $\boldsymbol{X}_i$ and labeled as l_i. All the indices of the target neighbors are contained in $\boldsymbol{\mathcal{S}}_i$. The objective of the second term is to push impostors of $\boldsymbol{X}_i$ (i.e. $\boldsymbol{Z}_r$) "far away", where $\boldsymbol{\mathcal{V}}_i = \{(j, r)|j \in \boldsymbol{\mathcal{S}}_i \text{ and } l_r \neq l_i\}$, $[x]_+ = max(x, 0)$. The trade-off between the pull and push terms is determined by $\mu \in [0, 1]$. The LMNN function was selected due to its robustness. Other loss functions could have been selected instead. The distance used in Equation (3.6) is the Mahalanobis distance between restricted affine hulls, specifically,

$$\begin{aligned}&\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) = (\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j)^T\boldsymbol{M}(\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j) \\ &(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j) = arg\min_{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_j}\left[(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)^T\boldsymbol{M}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)\right] \\ &s.t.\ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1,\ \|\boldsymbol{\beta}_j\|_{l_p} < \sigma_2, \sum_{n=1}^{N_i}\alpha_n^i = 1, \sum_{k=1}^{K}\beta_k^j = 1.\end{aligned} \tag{3.7}$$

However, Mahalanobis distance under other hull models [8, 33] can also be employed instead.

The prototype gallery $\mathbb{Z}$ and the Mahalanobis matrix $\boldsymbol{M}$ are optimized via solving Equation (3.7) in an EM-like manner. Specifically, gradient descent is employed to update $\mathbb{Z}$ and $\boldsymbol{M}$ in an alternating manner.

$\boldsymbol{M}$ Step: In this step, $\boldsymbol{M}$ is updated using gradient descent with the prototype $\mathbb{Z}$ fixed. At the $(t+1)^{th}$ iteration, the $\boldsymbol{M}$ is then updated via

$$\boldsymbol{M}^{t+1} = \boldsymbol{M}^{t} - \eta_{\boldsymbol{M}} \frac{\partial \mathcal{L}^{t}}{\partial \boldsymbol{M}^{t}}, \tag{3.8}$$

where $\eta_{\boldsymbol{M}}$ is the learning rate. The partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{M}$ is given by:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{M}} &= \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{M}} \\ &= \sum_{\boldsymbol{\mathcal{G}}} \left[(1-\mu) \sum_{\boldsymbol{\mathcal{S}}_i} \boldsymbol{C}ij + \mu \sum_{\boldsymbol{\mathcal{V}}_{i+}} (2\boldsymbol{C}ij - \boldsymbol{C}ir) \right], \end{aligned} \tag{3.9}$$

where,

$$\begin{aligned} \boldsymbol{C}ij =& (\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j)(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j)^T \\ \boldsymbol{\mathcal{V}}_{i+} =& \left\{ (j,r) \mid 2\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) - \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_r) > 0 \right\}. \end{aligned} \tag{3.10}$$

The representation coefficients $(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j)$ are calculated from Equation (3.7), and $\boldsymbol{\mathcal{V}}_{i+}$ is a subset of $\boldsymbol{\mathcal{V}}_i$, containing the index pairs (j,r) for which the hinge loss in $\mathcal{L}_i$ is larger than zero. To ensure that $\boldsymbol{M}$ is positive semi-definite, the updated $\boldsymbol{M}$ is projected onto its nearest positive semi-definite matrices as described in [32].

$\mathbb{Z}$ Step: In $(t+1)^{th}$ iteration, each prototype set $\boldsymbol{Z}_k^{t+1} \in \mathbb{Z}^{t+1}$ is optimized independently by:

$$\boldsymbol{Z}_k^{t+1} = \boldsymbol{Z}_k^{t} - \eta_{\mathbb{Z}} \frac{\partial \mathcal{L}^{t}}{\partial \boldsymbol{Z}_k}, \tag{3.11}$$

where $\eta_{\mathbb{Z}}$ is the learning rate for $\mathbb{Z}$. The partial derivative of the loss function $\mathcal{L}$ with respect to $\boldsymbol{Z}_k$

is the summation of partial derivative of loss on each gallery set:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_k} = \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial \mathcal{L} i}{\partial \boldsymbol{Z}_k}. \tag{3.12}$$

Since $\boldsymbol{Z}_k$ is considered to be a target neighbor for some of the gallery sets $\boldsymbol{X}_i$, but an impostor for a different $\boldsymbol{X}_i$, the corresponding partial derivatives vary. Specifically, when $\boldsymbol{Z}_k$ is treated as a target neighbor (i.e. $k \in \boldsymbol{\mathcal{S}}_i \text{ and } (k,l) \in \boldsymbol{\mathcal{V}}_{i+}$), its partial derivative is given by:

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{Z}_k} = & -2(1-\mu) \sum_{k \in \boldsymbol{\mathcal{S}}_i} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T \\ & - 4\mu \sum_{(k,l) \in \boldsymbol{\mathcal{V}}_{i+}} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T. \end{aligned} \tag{3.13}$$

When $\boldsymbol{Z}_k$ is treated as an impostor that violates the predefined margin (i.e. $(j,k) \in \boldsymbol{\mathcal{V}}_{i+}$), its partial derivative is given by:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_k} = 2\mu \sum_{(j,k) \in \boldsymbol{\mathcal{V}}_{i+}} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T. \tag{3.14}$$

In all other cases,

$$\frac{\partial \mathcal{L} i}{\partial \boldsymbol{Z}_k} = 0. \tag{3.15}$$

Update Neighborhood: Once $\mathbb{Z}$ or $\boldsymbol{M}$ has been updated, the corresponding distance and neighborhood relationship should be refined. The Mahalanobis metric $\boldsymbol{M}$ can be decomposed via Cholesky decomposition $\boldsymbol{M} = \boldsymbol{L}^T \boldsymbol{L}$. The distance between $\boldsymbol{X}_i$ and $\boldsymbol{Z}_j$ in Equation (3.7) can be written as:

$$\begin{aligned} & \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) = \left[\boldsymbol{L}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j)\right]^T \boldsymbol{L}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j) \\ & (\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j) = arg \min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j} \|\boldsymbol{L}(\boldsymbol{X}_i \boldsymbol{\alpha}_i - \boldsymbol{Z}_j \boldsymbol{\beta}_j)\|_2^2 \\ & s.t.\ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1,\ \|\boldsymbol{\beta}_j\|_{l_p} < \sigma_2. \end{aligned} \tag{3.16}$$

It is equivalent to first project $\boldsymbol{X}_i$ and $\boldsymbol{Z}_j$ into a new space defined by $\boldsymbol{L}$, and then calculates the geodesic distance between the two hulls in the projected space. Using the same trick in Equation (3.3), Equation (3.16) can be formulated as

$$\begin{aligned} &\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) \\ &= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j} \left(\|\mathbf{u}' - \hat{\boldsymbol{X}}_i' \boldsymbol{\alpha}_i - \hat{\boldsymbol{Z}}_j' \boldsymbol{\beta}_j\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}_i\|_2^2 + \lambda_2 \|\boldsymbol{\beta}_j\|_2^2 \right), \end{aligned} \tag{3.17}$$

where $\mathbf{u} = [\, \mathbf{0}\,;\, \mathbf{1}\,;\, \mathbf{1}\,]$, $\hat{\boldsymbol{X}}_i' = [\, \boldsymbol{L}\boldsymbol{X}_i\,;\, \mathbf{1}^T\,;\, \mathbf{0}^T\,]$, and $\hat{\boldsymbol{Z}}_j' = \left[-\boldsymbol{L}\boldsymbol{Z}_j; \mathbf{0}^T; \mathbf{1}^T\right]$. The column vectors $\mathbf{0}$ and $\mathbf{1}$ have the appropriate sizes associated with their corresponding contexts. Equation (3.17) has a closed-form solution. It can also be solved using the fast solver in [86] to update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j)$. Once $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \mathbb{Z})$ has been updated, the neighborhood relationships $(\boldsymbol{\mathcal{S}}_i, \boldsymbol{\mathcal{V}}_i)$ can be refined accordingly.

3.2 Implementation Details

Training: The training process is conducted to optimize prototype $\mathbb{Z}$ and the Mahalanobis matrix $\boldsymbol{M}$. An overview of the training procedure is offered by Algorithm 4.1.

Line 1 (Initialization): The matrix $\boldsymbol{M}$ is initialized using an identity matrix of the corresponding dimensions. The prototypes can be initialized in many ways, such as clustering or random sampling in the original image set. The values of the initial learning rates $\eta_{\boldsymbol{M}}$ and $\eta_{\mathbb{Z}}$ are set empirically.

Line 3 (Convergence criteria): In the implementation, the stopping condition is defined as the union of three criteria: (i) the relative change of $\mathcal{L}$ is smaller than a threshold $\omega_{\mathcal{L}}$ using a window of five iterations; (ii) both learning rates are smaller than a threshold ω_η; or (iii) there are no impostors.

Lines 4-6, 8, 10-12, 14 (Adaptive Learning rate): If the update overshoots (i.e. $\mathcal{L}^{t+1} > \mathcal{L}^t$), the learning rate was reduced by a factor of σ_r to increase the stability of the algorithm. If $\boldsymbol{M}$ and $\mathbb{Z}$

Algorithm 3.1: Set-based Prototypes and Metric Learning

input : $\mathcal{G}$
output: $\mathbb{Z}$, $\boldsymbol{M}$

Initialize $\boldsymbol{M}_0$, $\mathbb{Z}_0$, $\eta_{\boldsymbol{M}}$, $\eta_{\mathbb{Z}}$;
$(\mathbb{Z}, \boldsymbol{M})$ =*SPML* $\mathcal{G}$;
while *convergence criterion is not met* **do**
  while $\mathcal{L}^{t+1} > \mathcal{L}^t$ **do**
    $\eta_{\mathbb{Z}} = (1 - \sigma_r)\eta_{\mathbb{Z}}$;
  end
  Update $\mathbb{Z}$ (Equation (3.12));
  $\eta_{\mathbb{Z}} = (1 + \sigma_g)\eta_{\mathbb{Z}}$;
  Update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \mathbb{Z})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Equation (3.17));
  while $\mathcal{L}^{t+1} > \mathcal{L}^t$ **do**
    $\eta_{\boldsymbol{M}} = (1 - \sigma_r)\eta_{\boldsymbol{M}}$;
  end
  Update $\boldsymbol{M}$ (Equation (3.8));
  $\eta_{\boldsymbol{M}} = (1 + \sigma_g)\eta_{\boldsymbol{M}}$;
  Update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \mathbb{Z})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Equation (3.17));
end

are updated successfully, the corresponding learning rates were increased by a factor σ_g to speed up the convergence. The values of σ_r and σ_g were set empirically.

Testing: In the testing, a probe image set $\boldsymbol{P}$ was compared with all the prototype sets in the gallery, and their distances $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{P}, \boldsymbol{Z}_i), i \in [1, N]$ were calculated. The probe was classified to

$$y_i = arg \min_i \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{P}, \boldsymbol{Z}_i), \tag{3.18}$$

the same subject with its closest prototype gallery set.

Discussion: The testing time complexity of the proposed SPML with its most related hull based algorithms: SSDML and RNP were compared. The elementary operation of the testing process (Equation (3.18)) was calculating the distance between a gallery set and a probe set.

RNP: There were two ways to compute the distance between two sets (Equation (3.3)): (i) a

closed-form solution, and (ii) a fast solver. For the closed-form solution, the time complexity is $\mathcal{O}\left((N_i + N_q)^3\right)$, where N_i and N_q denote the number of images in the current gallery set and probe image set, respectively. For the alternate fast solver, the time complexity is $\mathcal{O}\left(dT(N_i + N_q)\right)$, where T is the number of iterations and d is the feature dimensionality.

SSDML: An extra step of mapping features to the learned space was added to RNP. Its time complexity was $\mathcal{O}\left(d^3(N_i + N_q)\right)$. The overall time complexity was $\mathcal{O}\left((N_i + N_q)^3\right)$ for the closed-form solution and $\mathcal{O}\left((d^3 + dT)(N_i + N_q)\right)$ for the fast solver.

SPML: The distance between two sets was computed in the same manner with SSDML, by replacing the gallery set with the prototype set. The overall time complexity was reduce to $\mathcal{O}\left((K + N_q)^3\right)$ and $\mathcal{O}\left((d^3 + dT)(K + N_q)\right)$ for the closed-form solution and the fast solver, respectively, where the number of prototypes per set K is smaller than N_i. Based on the analysis above, the testing time was increasing linearly or in a cubic manner with the number of images per gallery set. Representing the gallery image set using fewer samples will reduce the testing time significantly.

3.3 Experiments

In this section, the proposed SPML was evaluated and compared with state-of-the-art algorithms. In this section, experiments were designed to evaluate the proposed SPML framework. The state-of-the-art algorithms from each category were selected as baselines, namely CDL [71], GDL [26], RNP [86], SSDML[89], ISCRC [90].

Table 3.1: A summary of the datasets used in the experiments.

Dataset	Subjects	Image Sets per Subjects in Gallery	Probes	Images per Sets
ETH-80	8	5	40	41
YTC	47	5	1,621	13 − 349
YTF	59	4	67	48 − 2,157

Numbers listed in the table are computed based on the protocol and processing. There are some differences from the statistics of the original release that are explained in Section 3.3.1.

3.3.1 Datasets

The ETH-80 [44], YouTube Celebrity (YTC) [40] and YouTube Face (YTF) [80] datasets were selected to assess the performance of the proposed SPML in object categorization and video-based face identification. The basic information about the employed datasets is summarized in Table 3.1.

ETH-80: This dataset comprises objects from eight categories, where each category contains 10 objects. For each object, 41 images from different views are captured to form an image set. Following [71], the original images are resized to 20×20 and the concatenated pixel values were used as features. In all the experiments, five objects were randomly sampled from each category to form the gallery, while the rest were used as probes. The random splitting of gallery and probe were repeated ten times, and the average performance was reported.

YTC: This dataset contains low resolution video sequences of 47 subjects from YouTube. For each subject, the number of videos varies from 15 to 106. Following [89], the face area was detected frame by frame, resized to 30×30, and the concatenated pixel values were used as features. For each video, the number of valid image frames (i.e. a face is detected) varies from 13 to 349. In all experiments, four videos were randomly sampled from each subject to form the gallery, while the rest were used as probes. The random splitting of gallery and probe were repeated ten times, and the average performance was reported.

YTF: This dataset contains 3,425 videos captured from 1,595 different subjects. The objective was to simulate a face identification task. However, for most subjects only a single video was available. As a result, these videos could not be used to evaluate the identification performance. To this end, a subset of 59 subjects was selected for which five or more videos were available. For each video, the number of valid image frames (i.e. a face is detected) varied from 48 to 2,157. This dataset comes with three feature descriptors: Local Binary Patterns (LBP), Center-Symmetric LBP (CSLBP), and Four-Patch LBP (FPLBP). In all the experiments, four videos are randomly sampled from each subject to form the gallery, while the rest are used as probes. The random splitting of gallery and probe are repeated for ten times and report the average performance.

Feature Processing: Principal Component Analysis (PCA) is applied to all the features for two reasons: (i) to reduce the noise in the features; and (ii) to avoid the over-fitting introduced by high dimensional features. Except for RNP, all the algorithms evaluated used either a distance metric learning (i.e. CDL, SSDML, and SPML) or dictionary learning (i.e. GDL, and ISCRC). Moutafis et al. [57] illustrated experimentally that distance metric learning algorithms suffer from overfitting when high-dimensional features were used, because the parameters that needed to learn increased quadratically with the feature dimensionality. Similar reasons also apply to dictionary learning algorithms. In particular, the length of the feature vectors is arbitrarily set to 100 as default in the experiments. A sensitivity analysis on the feature length is presented in Experiment 2. To reduce large intra-class variations, these PCA reduced features were projected onto an intra-class subspace following the procedure described in [7].

3.3.2 Baselines

In this section, the algorithms compared in the experiments and their corresponding parameter settings are discussed. To conduct a fair comparison, all parameters are tuned according to the

Table 3.2: A summary of the algorithms compared in the experiments.

Algorithm	Literature Source	Category
CDL [71]	Wang *et al. CVPR' 12*	Statistical
GDL [26]	Harandi *et al. ICCV' 13*	Subspace
RNP [86]	Yang *et al. FG' 13*	Hull
SSDML [89]	Zhu *et al. ICCV' 13*	Hull
ISCRC [90]	Zhu *et al. TIFS' 14*	Hull

instructions in the original papers. In particular, the gallery set was split into gallery and validation via random sampling one image set per subject/object. All the tuning was conducted to achieve the highest identification rate in the validation set. All the parameters were initialized using the default values. All the changes are described in the following paragraphs. A summary of the selected algorithms is provided in Table 3.2. In particular, recently published algorithms from the three categories discussed in Chapter 2 were selected.

Statistical model: Covariance Discriminative Learning (CDL) [71] was selected due to its stable performance reported in the literature. In particular, it represents each image set using its covariance matrix, and the set-to-set distance is calculated on the Riemannian manifold. Two versions of implementations based on Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS) were provided by Wang *et al.* [74], referred to as CDL-lda and CDL-pls. The parameters for this method were kept the same as default settings in the code after cross-validation.

Subspace model: Grassmann dictionary learning (GDL) [26] is one of the most recent algorithms that uses the linear subspace model. In particular, it extends the dictionary learning and sparse coding into the subspace model. The implementation is provided by Harandi *et al.* online [23]. In the experiments, the orthogonal representation of linear subspace is computed using Singular Value Decomposition. In each experiment, the order of the subspace was kept the same as the number of

prototypes used. The number of atoms in the dictionary was set to 20, 150, and 232 for ETH-80, YTC, and YTF, respectively. Numbers were selected based on cross-validation. The rest of the parameters were kept the same as the default settings after cross-validation.

Hull model: Except for RNP and SSDML (introduced in Chapter 2), the image set based collaborative representation and classification (ISCRC) [90] method was also included. As RNP was used to model the distance for SSDML, ISCRC, and the proposed SPML, its performance was used as a baseline in the experiments. The implementations of all above algorithms was provided by Zhu *et al.* online [88]. For RNP the regularization parameters λ_1 and λ_2 are set to 10 based on the result of cross-validation. A nearest neighbor classifier is used for testing. For SSDML, the numbers of similar and dissimilar sets were set to be three and 30, accordingly.

SPML: For the proposed algorithm, the default settings were provided here. Each prototype set was initialized using k-means clustering, while the number of prototypes used was set to 10. As an LMNN-like objective function was employed, the related parameters were set to follow the original implementation of LMNN. The trade-off parameter μ (Equation (3.6)) was set to 0.5 to equally weight the "pull" and "push" terms. The convergence threshold $\omega_{\mathcal{L}}$ was set to 0.01. The learning rates $\eta_{\boldsymbol{M}}$ and $\eta_{\mathbb{Z}}$ were initialized to 0.01. The learning rate threshold ω_η was set to 10^{-7}. The number of neighbors was set to three. The growth and reduction rates σ_g and σ_r (see Algorithm 4.1) were set to 0.05 and 0.5, respectively. Following the settings of RNP, the regularization parameters λ_1 and λ_2 (Equation (3.17)) were set to 10.

3.3.3 Experiments and Results

Experiment 1: The objective of this experiment was to compare the classification performance of SPML with state-of-the-art approaches. The number of prototypes used for gallery sets was set to 10. To reduce the computational cost on YTF the samples per set in the probe were reduced to 100

Table 3.3: A summary of the results for Experiment 1.

Method	ETH-80		YTC		YTF					
					LBP		FPLBP		CSLBP	
	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
CDL-lda	82.50	6.22	60.07	4.69	36.27	4.28	31.34	2.49	35.37	4.63
CDL-pls	80.75	8.07	60.14	2.96	37.31	5.04	30.90	3.89	35.22	5.43
GDL	82.50	5.47	**64.82**	3.09	51.34	5.13	46.57	5.20	50.03	4.81
RNP	78.75	5.51	64.11	2.11	52.24	3.20	**53.13**	7.10	45.52	4.63
SSDML	83.25	5.71	61.63	2.51	56.42	5.16	52.39	3.43	51.79	4.11
ISCRC	67.50	4.74	47.84	2.71	52.24	4.69	48.21	3.60	39.40	3.80
SPML	**85.75**	6.43	61.67	3.00	**62.39**	6.11	50.90	4.40	**52.54**	2.56
p value	0.1065		**0.0026**		**0.0024**		0.3171		0.0024	

The values denote the mean (%) and standard deviation (%) of rank-1 identification rate when only 10 prototypes are used. The last row shows the results of the 10-fold cv paired t-test between SPML and the baseline with the highest performance.

using k-means clustering. For ETH-80 and YTC, the original probe sets were used. An overview of the results is offered in Table 3.3. As illustrated, SPML appears to outperform all methods for ETH-80, and two out of three features for YTF. The statistic-based CDL needs enough samples to estimate the covariance matrix. The subspace-based GDL can embed the information into a low order subspace. In the order of 10, it seems to perform worse than the proposed SPML. The RNP learned an unsupervised distance and thus did not fully utilize the labels of the training data. SSDML learned a distance metric, but the reduction in the number of samples per set appeared to degrade its performance. ISCRC used dictionary learning to compress the image set. However, the fitting does not appear to work well. Finally, SPML utilizes the training data more effectively to compress the input information into fewer prototypes. The inherent ability to perform a reduction in the number of samples per gallery set gives the edge to SPML over other methods. To verify

whether the performance differences are statistically significant between SPML and other methods, a 10-fold cv paired t-test was conducted. In particular, the performance between SPML and the baseline with the highest performance was compared. Corresponding p values were reported in Table 3.3. At a significant level of 0.05, SPML achieves significantly better performance for LBP features in YTF. The performance is comparable with the best baseline in ETH-80, LBP and CSLBP in YTF. SPML get worse performance in YTC dataset, and fails to improve the baseline provided by RNP. Further analysis is provided in Experiment 2 and Experiment 6.

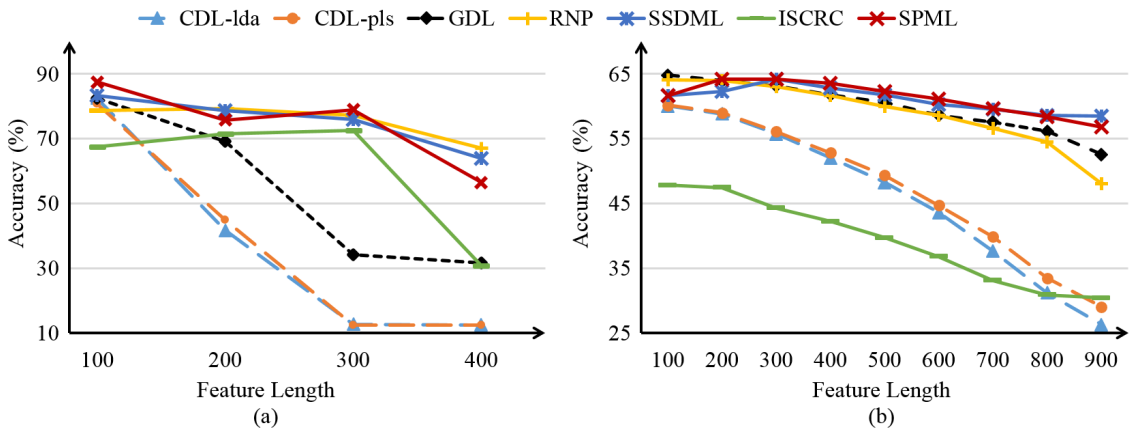

Figure 3.2: Average rank-1 identification accuracy obtained using different length of features (Experiment 2): (a) results obtained in the ETH-80 dataset; (b) results obtained in the YTC dataset. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. Results for the subspace-based algorithms are presented in long dashed lines. The feature length covers from 100 to the original length without reduction with a step of 100.

Experiment 2: The objective of this experiment was to assess the impact of the feature length on the identification performance. An overview of the results was depicted in Figure 3.2. The ETH-80 dataset and YTC dataset were used to assess the performance. The feature length used ranges from 100 to the original length, with a step of 100. In ETH-80, the performance of all algorithms was decreasing with the increase of the feature length. There were several reasons: (i) longer feature

contains more noise; (ii) longer feature corresponds to more parameters to learn for each model. It may under-fit due to lack of training data or over-fit due to the complex model. It was also observed that the hull-based methods (i.e. solid lines) were more robust to the changes of feature length. The statistic-based methods (i.e. long dashed lines) suffered the most from high dimensional features. Similar patterns were observed from the results obtained in YTC. The performance of SPML and SSDML started to decrease after the feature length of 300. With a feature length of 200 and 300, SPML achieved comparable performance with the highest one.

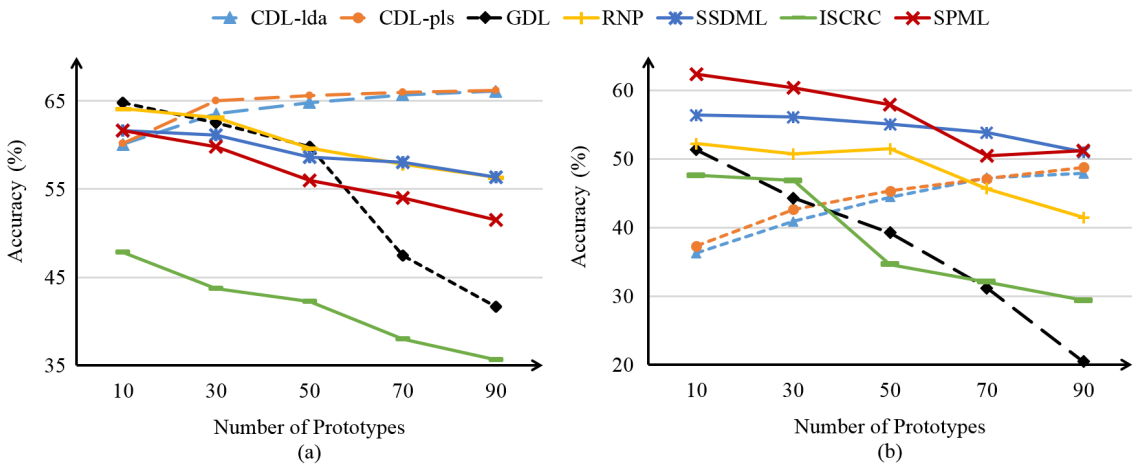

Figure 3.3: Average rank-1 identification accuracy obtained using different numbers of prototypes (Experiment 3): (a) results obtained in the YTC dataset; (b) results obtained in the YTF dataset using LBP features. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. Results for the subspace-based algorithms are presented in long dashed lines.

Experiment 3: The objective of this experiment was to assess the impact of the number of prototypes used on the identification performance. An overview of the results was depicted in Figure 3.3. The results were obtained in YTC and YTF datasets, because they contain more subjects and provide larger image sets. The number of prototypes used ranged from 10 to 90 with a step of 20. All other settings were kept the same with Experiment 1. In YTF, the LBP feature was used

because it yielded the best accuracy in Experiment 1 for six out of seven algorithms. Some image sets contained fewer samples than the target number of prototypes to be learned. In such cases, the number of prototypes was set to the number of samples in the original set. In general, the SPML achieved the highest performance using only 10 prototypes. This indicates its effectiveness in compressing the available information using few prototypes. In YTF, the SPML appears to outperform all other methods in four out of five cases. In YTC, the performance of SPML is on average, when fewer prototypes are used. Better performance is expected with longer features as illustrated in Experiment 2. The performance of CDL (short dashed lines in Figure 3.3) increased with an increase of the number of prototypes used, while the performance of all other algorithms is decreased. Since CDL relies on a statistical model, it needs enough samples to estimate the covariance matrix. The subspace-based GDL (the long dashed line in Figure 3.3) was not suitable to represent an image set using a high order subspace. For hull-based SPML, ISCRC, SSDML, and RNP (solid lines in Figure 3.3), spanning a large amount of vectors caused an overlap of inter-class hulls. It may also result in over-fitting for SPML when a large number of prototypes was used.

Experiment 4: The objective of this experiment was to assess the impact of different numbers of subjects in the gallery. The cumulative match characteristic curve (CMC) was employed to assess the performance. The gallery set was expanded by adding some of the removed subjects which contained four videos. All these samples were added to the gallery set without matching samples in the query. This was due to the restriction that three neighbors were needed in the training process. Therefore, four videos per subject should ensure in the gallery for training. A summary of the results is depicted in Figure 3.4. The CDL-pls and ISCRC are not applicable for CMC. Their classification is based on all samples from a certain class instead of a single sample. As indicated, the performance of all algorithms droped after expanding the gallery set. It increases the possibility of a wrong match. Note that the proposed SPML could only outperform RNP and SSDML before rank 3. One of the reason is that only the three nearest neighbors were considered in the objective

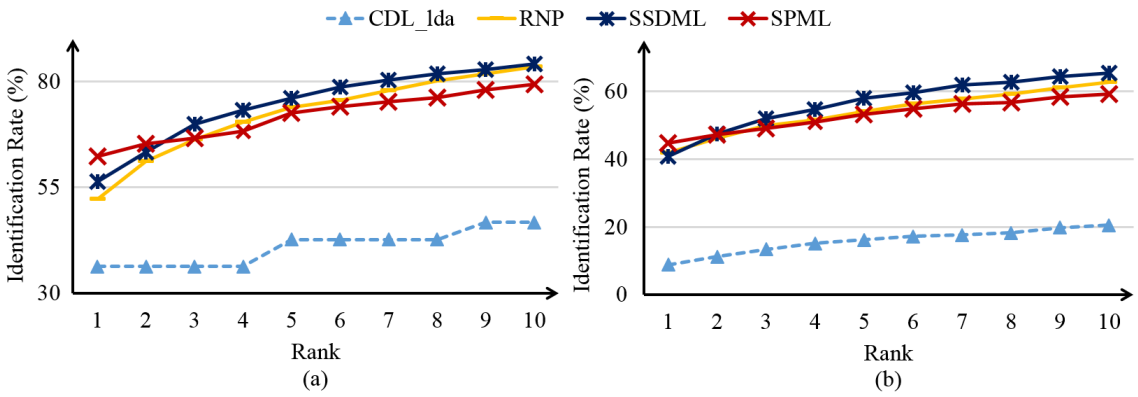

Figure 3.4: The CMC curves obtained with different numbers of subjects in gallery (Experiment 4): (a) results obtained in the default settings of YTF; (b) results obtained with the expanded gallery set. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. The classification of CDL-pls and ISCRC are not applicable for the CMC curves.

functions. As a result, only the first three recalls were optimized. This is one of the limitations of the proposed algorithm. This limitation can be addressed by embedding other distance metric learning objective functions without the local neighborhood constraints.

Experiment 5: The objective of this experiment was to assess the impact of different initialization approaches on the identification accuracy. In particular, Two initialization approaches: (i) k-means clustering and (ii) random sampling are compared. In particular, for each gallery-probe split, the algorithm was tested using 30 different random initialization settings. This experiment was conducted using the ETH-80 dataset, setting the number of prototypes to 10 (as in Experiment 1). A summary of the results is depicted in Figure 3.5. The blue star-symbol denotes the rank-1 identification rate obtained using k-means initialization. The red plus-symbol denotes the extreme results treated as outliers. As indicated, in every split, the performance of random initialization can be worse or better than k-means initialization. However, in five out of 10 cases the performance of k-means initialization was better than the average performance of random sampling; and in four

out of 10 cases they were comparable. This indicates that k-means clustering offers a better initialization for the proposed SPML. This is expected as the random sampling strategy results in loss of important information.

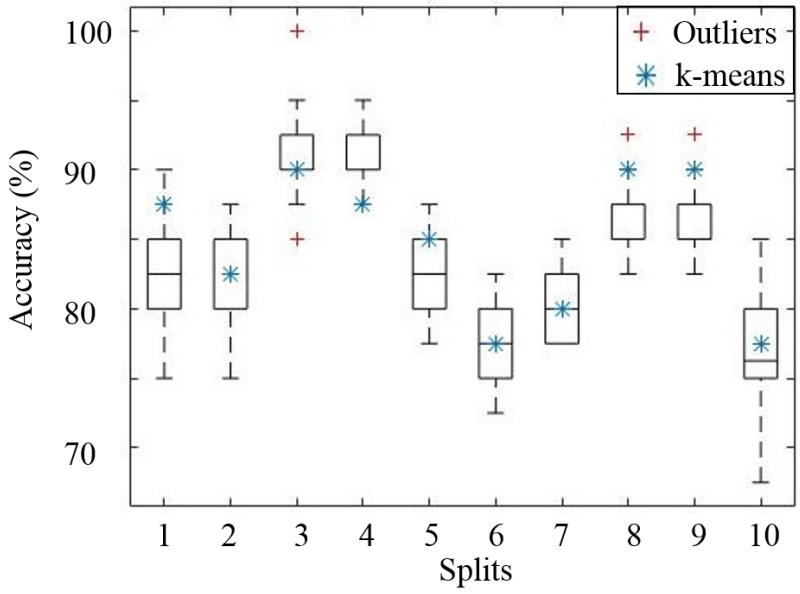

Figure 3.5: Depicted are boxplots for the rank-1 identification rate obtained in different splits of gallery-probe settings (Experiment 5). In each split, the accuracy is computed 30 times using different random initialization. The red plus-symbol denotes for the outlier results in random initialization, and the blue star-symbol denotes the rank-1 identification rate obtained using k-means initialization.

Experiment 6: The objective of this experiment was to analyze the failure case (i.e. using FPLBP features in TYF with 10 prototypes) reported in Experiment 1. In particular, the objective value, training and testing accuracy on each iteration were analyzed. Results are summarized in Figure 3.6 (a). The performance obtained using LBP in YTF dataset is provided in Figure 3.6 (b) for comparison. As illustrated, in both cases the objective value and training accuracy converged very fast. For the FPLBP feature, its testing accuracy decreased with the updates. For the LBP feature,

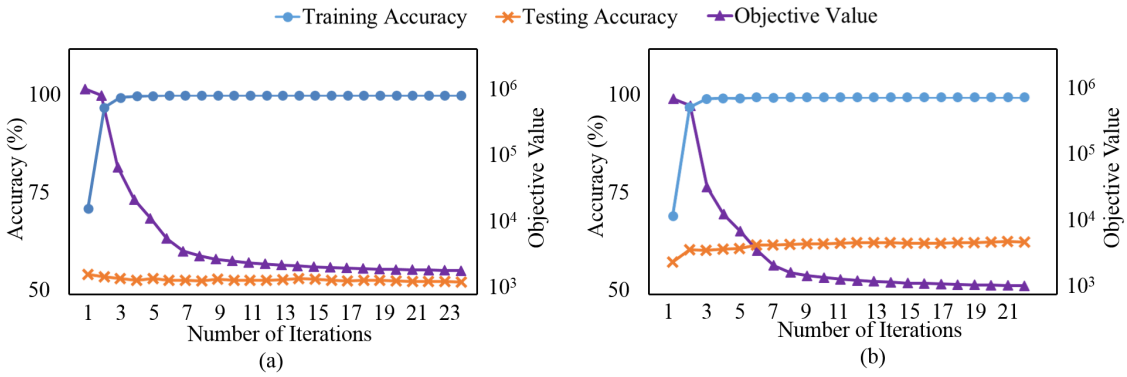

Figure 3.6: Convergence property of objective value, training and testing accuracy (Experiment 6): (a) results obtained in YTF using FPLBP features; (b) results obtained in YTF using LBP features. In both figures, the left axis indicates accuracy (training and testing); the right axis indicates the objective value. All numbers reported are the average values.

its testing accuracy increased slowly with the updates, and finally achieved a 5% improvement. One possible reason is that the learning process overfits from the first iteration. To verify this interpretation, the learning rate of both prototype learning and metric learning (i.e. $\eta_{\boldsymbol{M}}$ and $\eta_{\mathbb{Z}}$) were decreased to 0.001, while keeping all other settings fixed. The obtained results are depicted in Figure 3.7. As illustrated, the testing accuracy improved in the first five iterations, and decreased from the sixth iteration. This observation verified the interpretation that the failure was caused by overfitting from the first iteration in the failure case ($\eta_{\boldsymbol{M}}, \eta_{\mathbb{Z}} = 0.01$). This overfitting was caused by the large initial learning rate. Although the overfitting at the first iteration can be avoided by decreasing the initial learning rate, it still suffers from overfitting after the convergence (Figure 3.7). The small learning rate will also result in a long convergence time. A practical way to address this problem is to split a validation set from gallery to cross-validate a proper initial learning rate and a stop criterion.

Experiment 7: The objective of this experiment was to analyze the contributions of two learning procedures in SPML: set-based metric learning (SML) and set-based prototype learning (SPL). In

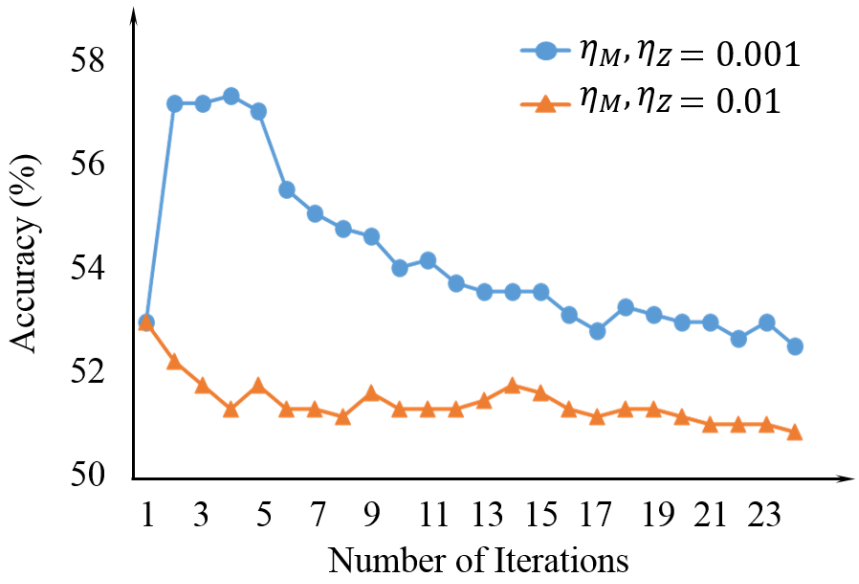

Figure 3.7: Convergence property of testing accuracy (Experiment 6) obtained using a reduced learning rate.

SML, the results of k-means initialization were used as fixed prototypes. The Mahalanobis metric was learned by minimizing Equation 3.12. The updating rules were kept the same as described in Equation 3.8. In SPL, the Mahalanobis M was fixed as an identity matrix. The prototypes were learned by minimizing Equation 3.12. The corresponding updating rules were kept the same as described in Equation 3.8. In particular, the testing accuracy of SPML, SML, and SPL at each iteration were reported. Corresponding results are summarized in Figure 3.8. Results were obtained using LBP features in YTF datasets. All the settings were kept the same as Experiment 1. As illustrated, the SPL converged very fast (only 13 iterations). However, the testing accuracy did not fit well with the prototype learning procedure. The testing accuracy kept decreasing after the second iteration. The SML was much more stable than the SPL. The testing accuracy increased in the first few iterations and then started to decrease. The combined SPML, on the other hand, kept increasing the testing accuracy on each update. As a result, there was no single step that made

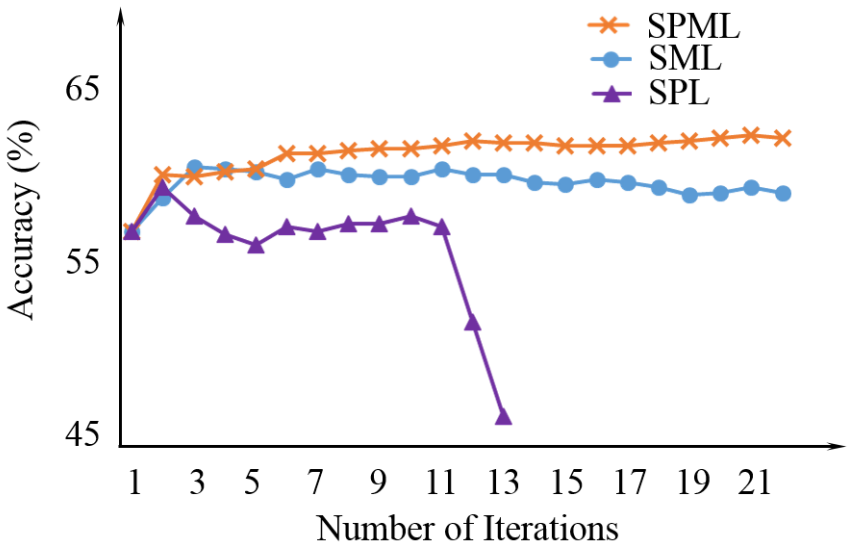

Figure 3.8: Convergence property of testing accuracy (Experiment 7) obtained using SPML, SML, and SPL. Results are obtained using LBP features in YTF.

SPML work. Both the prototype learning and metric learning process contributed to performance of SPML.

Experiment 8: The objective of this experiment was to access the impact of outliers on set-to-set identification performance. Following [8], three protocols are discussed: (i) outliers in gallery set, outliers in prob set, and outliers in both. For all the protocols, the 5% of outlier images were added to corresponding image sets. These outliers were randomly sampled from other classes. This experiment was conducted on ETH-80 with the number of prototypes set to ten. The results are illustrated in Table 3.4. It appears that the proposed SPML is more robust to outliers.

Table 3.4: Summary of results for Experiment 8. The values denote the mean(%) average performance drop of rank-1 identification compared with clean data.

Method	Performance Change Compared with Clean Data (%)		
	Outliers in Gallery	Outliers in Prob	Outliers in Both
CDL-lda	-0.79	-3.73	-2.55
CDL-pls	-0.56	-1.69	-1.38
GDL	-0.32	-1.53	-1.37
RNP	-3.48	-5.19	-5.37
SSDML	-2.58	-1.73	-1.89
ISCRC	5.24	-10.40	-17.59
SPML	**0**	**-1.37**	**-1.37**

Chapter 4

Objective 2: Common Templates for Point-to-Set Matching

In this chapter, the objective was to develop an algorithm that generated a common template for point-to-set matching. In particular, the attention-based model were employed to take advantage of the recent advances in deep learning architectures. As described in Chapter 2, Section 2.1.4, the key issue in the attention-based module was to quantify the discriminative information level of each sample in a set. To this end, a confidence-driven network (CDN) was developed to learn a confidence-based score distribution within an image set. The learned confidence scores can (i) represent the discriminative information level of each sample in set, and (ii) help improve the decision aggregation for point-to-set matching. An overview of its key characteristics is illustrated in Figure 4.1. It comprises two parts: (i) a feature extraction network (FEN), and (ii) a performance prediction network (PPN). FEN is a distance-based ConvNet architecture, and a pre-trained point-to-point network can be employed. PPN is a binary classification network. It takes as an input an intermediate feature representation obtained from FEN and generated a confidence score.

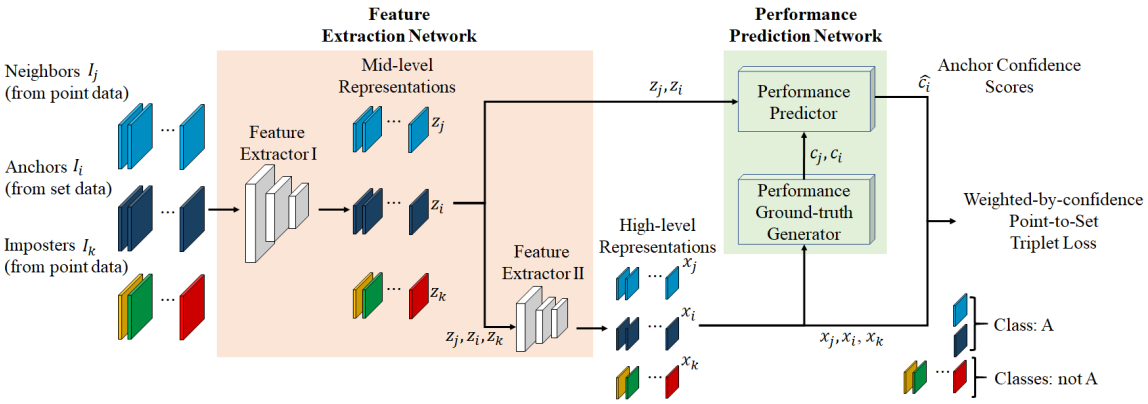

Figure 4.1: The proposed CDN is trained on triplet batches $(\boldsymbol{I}_i, \boldsymbol{I}_j, \boldsymbol{I}_k)$. In particular, it contains two parts. The first part is a feature extraction network. The second part is a performance prediction network, which comprises a ground-truth generator and a performance predictor. The high-level features that contain information about the identity are leveraged to generate "ground-truth" target estimations for the given batch of triplets using a single-sample test discussed in Section 4.1.2.1. The middle-level features along with the generated targets are used to train the performance predictor which generates confidence scores $\hat{c}_i$ for each of the anchors in the set. The network is trained jointly using the weighted-by-confidence point-to-set triplet loss introduced in Section 4.2.2.

This score indicates the probability that the extracted feature vector will contribute to a correct decision. To estimate the ground-truth confidence score, single-sample tests were proposed to generate ground-truth for the confidence score. In the training phase, the confidence score will guide the feature extraction network to put less "attention" on the samples with low confidence levels. It helps avoid overfitting on samples that are possibly difficult, or others that the model is uncertain about its predictions. In the matching phase, the confidence scores were used to fuse the results from different samples of the same set. In summary, the contributions of the proposed CDN are the following: (i) a weighted-by-confidence point-to-set triplet loss that enables us to adapt a point-to-point network to a point-to-set network; and (ii) a single-sample test mechanism to quantify the discriminative level of a sample.

4.1 Method

4.1.1 Feature Extraction Network

The objective was to leverage the discriminative power of feature representations in a point-to-point matching setup and adjust them in a point-to-set matching protocol, by introducing only a few changes to the original architecture. Thus, the center loss face ResNet [31] developed by Wen et al. [79] was selected as a base architecture, for its outstanding performance. To adapt FEN to point-to-set matching, a weighted-by-confidence point-to-set triplet loss was proposed. The triplet loss [60] was chosen because of its ability to learn discriminative feature representations that can be generalized to unseen classes. In particular, the training data were split into triplet batches

$$\mathcal{B} = \{(I_i, I_j, I_k)\,|, l_j = l_i = l_+, l_k \neq l_+\}\,, \tag{4.1}$$

with the restrictions: (i) all the anchor images, I_i, are sampled from an image set; (ii) the neighbors, I_j, and imposters, I_k, are sampled from the point data; (iii) the anchors and the neighbors are sampled from the same subject (i.e. $l_j = l_i = l_+$); and (iv) the anchors and the imposter are from different subjects (i.e. $l_k \neq l_+$). The weighted-by-confidence point-to-set triplet loss in a batch was formulated as follows:

$$\mathcal{L}_{\mathcal{B}} = \frac{\sum_{\mathcal{B}} \hat{c}_i \left[\mathcal{D}^2\left(\mathbf{x}_i, \mathbf{x}_j\right) - \mathcal{D}^2\left(\mathbf{x}_i, \mathbf{x}_k\right) + \alpha\right]_+}{\sum_{\mathcal{B}} \hat{c}_i}, \tag{4.2}$$

where $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ are the feature representations of the anchors, neighbors, and imposters, respectively. The objective of the first term is to the pull the neighbors $\mathbf{x}_j$ "closer", while the objective of the second term is to push the imposters $\mathbf{x}_k$ "far away", by a margin α. The proposed loss $\mathcal{L}_{\mathcal{B}}$ is the sum of the losses over all the triplets in the batch, weighted by the corresponding confidence scores $\hat{c}_i$ of the anchors $\mathbf{x}_i$ from an image set. The confidence score $\hat{c}_i$ generated by the performance predictor guides FEN to put less attention on the less informative samples.

4.1.2 Performance Predictor

The performance predictor is a binary classification network that distinguishes between informative and non-informative samples. In the implementation, a small ConvNet (just Conv-Pool-FC-FC) was employed to perform classification and used the softmax output probabilities as confidence scores. Its input are middle-level features from feature extractor I. The middle-level representation was used because it preserves more information from the original image, while in the high-level feature representations only the identity-related information was preserved.

4.1.2.1 Ground Truth Generator

In order to train the performance predictor in a supervised manner, "ground truth" confidence scores were necessary. However, such measurements are not available, and need to be generated. A conceptually straightforward approach was proposed to estimate the "ground truth" targets for the confidence scores.

The confidence score proposed in this chapter is the likelihood of a correct decision,

$$c_i = P(\hat{l}_i = l_i | \mathbf{x}_i), \tag{4.3}$$

where $\mathbf{x}_i$ is the anchor feature representation from Feature Extractor II, $\hat{l}_i$ is the rank-1 identity label returned from a distance based ranking, and l_i is the ground truth identity label. The confidence score c_i, indicates the probability of returning a correct decision (i.e. $\hat{l}_i = l_i$), based on the given feature representation $\mathbf{x}_i$.

To estimate this likelihood ground truth c_i, a single-sample test mechanism was proposed, which was performed during training within a triplet batch. An overview of this mechanism is provided in Figure 4.2. The input batch comprises B sets of triplets, including:

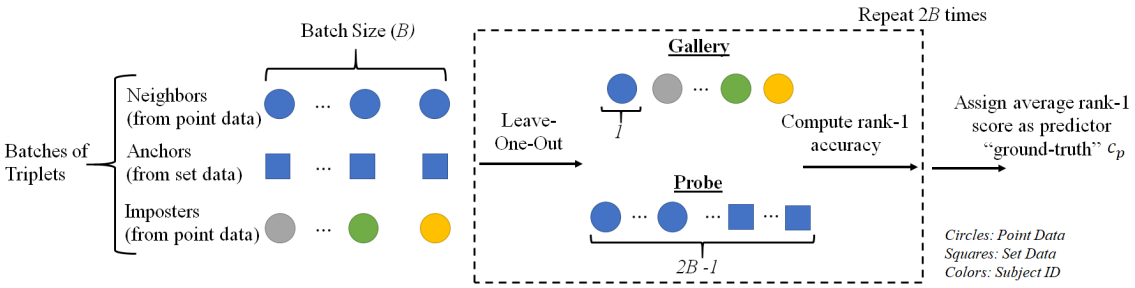

Figure 4.2: Single-sample test mechanism to generate ground truth targets for the performance predictor. Given high-level batches of triplets, a single sample from the neighbors or the anchors is put in the gallery along with all the imposters. The rest of the neighbors and the anchors form the probe. The rank accuracy is computed across 2B iterations.

(i) Anchors $\mathcal{A} = \{(\mathbf{x}_i, l_+) \mid i \in [1, B]\}$, from the set data;

(ii) Neighbors $\mathcal{A}_+ = \{(\mathbf{x}_j, l_+) \mid j \in [B+1, 2B]\}$, from the point data;

(iii) Imposters $\mathcal{A}_- = \{(\mathbf{x}_k, l_k) \mid k \in [2B+1, 3B]\}$, from the point data.

To better distinguish the anchors, the neighbors, and the imposters, their indices were defined from different ranges within the batch. The neighbors and the anchors were selected from the same subject l_+, whereas the negative samples were selected such that they are from B different subjects l_k. The single sample test was conducted via simulating an identification scenario, with the following steps.

Step 1: Gallery and Probe Enrollment. The gallery comprises all imposter samples, and one sample $\mathbf{x}_b$ from the union of the anchors and neighbors. Therefore, the gallery can be formulated as:

$$\mathcal{G}^b = \mathcal{A}_- \cup \{(\mathbf{x}_b, l_+)\}\,, \tag{4.4}$$

which contains one sample for every identity in the batch. Then the rest of the samples from the

anchor and the neighbor sets serves as a probe which is defined as:

$$\mathcal{P}^b = \mathcal{A} \cup \mathcal{A}_+ - \{(\mathbf{x}_b, l_+)\}\,, \tag{4.5}$$

where all probes were from the same identity l_+.

Step 2: Single Sample Test. Each probe sample $\mathbf{x}_p \in \mathcal{P}^b$ was compared with each gallery sample $\mathbf{x}_g$ by computing their Euclidean distance $d(\mathbf{x}_p, \mathbf{x}_g)$. Then the label for $\mathbf{x}_p$ was assigned to be the same with the identity in the gallery that has the smallest distance:

$$\hat{l}_p^b = l_{\hat{g}},\ \text{where}\ \hat{g} = \arg\min_g d(\mathbf{x}_p, \mathbf{x}_g). \tag{4.6}$$

The decision and the subject ID ground truth l_+ are compared, and the rank-1 hit for the sample $\mathbf{x}_p$ was computed as:

$$r_p^b = \begin{cases} 1 & \text{when} \quad \hat{l}_p^b = l_+ \\ 0 & \text{when} \quad \hat{l}_p^b \neq l_+ \end{cases}. \tag{4.7}$$

Step 3: Step 1 and Step 2 were repeated $2B$ times until each element in $\mathcal{A} \cup \mathcal{A}_+$ has enrolled in the gallery only once. The likelihood ground truth for each sample which has served as probe $\mathbf{x}_p$, was then computed as the average rank-1 hit rate across all testing iterations:

$$c_p = \frac{\sum_{b=1}^{2B} r_p^b}{2B - 1}\,, \tag{4.8}$$

where each sample from the anchors and the neighbors has served as a probe for $2B - 1$ times.

Note that, the likelihood "ground truth" is generated for both anchors and neighbors, despite using only the anchor scores in the weighted triplet loss in Equation (4.2). However, when training the performance predictor, both the neighbors and the anchors were involved to enlarge the training data.

Algorithm 4.2: Confidence-Driven Network

input : Batches of image triplets (I_j, I_i, I_k).

output: Network parameters θ_c, θ_p for classification f_c and performance prediction f_p, networks respectively.

1 Initialization: step $s = 0$, θ_c^s, θ_p^s;

2 **while** *(validation loss decreases)* **do**

3 $s \leftarrow s + 1$ $(\mathbf{z}_j, \mathbf{z}_i, \mathbf{z}_k) \leftarrow f_I(\mathbf{I}_j, \mathbf{I}_i, \mathbf{I}_k)$; `// Feed-forward to compute middle-level features from Feature Extractor I`

4 $\mathbf{x}^{s-1} = (\mathbf{x}_j^{s-1}, \mathbf{x}_i^{s-1}, \mathbf{x}_k^{s-1}) \leftarrow f_c(\mathbf{z}_j, \mathbf{z}_i, \mathbf{z}_k)$; `// Feed-forward to compute high-level features from Feature Extractor II`

5 $c^{s-1} = (c_j^{s-1}, c_i^{s-1}) \leftarrow SST(\mathbf{x}^{s-1})$; `// Single-sample test to estimate confidence ''ground truth''`

6 $\theta_p^s = SGD(\mathcal{L}_p \left(f_p((z_j, z_i), \theta_p^{s-1}), c^{s-1}\right))$; `// Update` θ_p

7 $\hat{c}_i^s = f_p(z_i, \theta_p^s)$ Feed-forward to compute confidence scores ;

8 $\theta_c^s = SGD(\mathcal{L}_\mathcal{B}\left(\mathbf{x}^{s-1}, \hat{c}_i^s, \theta_c^{s-1}\right))$; `// Update` θ_c `using Equation (4.2)`

9 **end**

4.2 Implementation Details

4.2.1 Training

During training, the Feature Extractor I remained frozen, whereas Feature Extractor II and PPN were trained jointly so as to enable information sharing through the weight updates. An overview is provided in Algorithm 5.1.

Lines 4-5: During training triplets of images were fed to the Feature Extractor I (denoted by f_I) to obtain intermediate features $(\mathbf{z}_j, \mathbf{z}_i, \mathbf{z}_k)$. These were then provided to the Feature Extractor II which outputs high-level representations $\mathbf{x}^{s-1} = (\mathbf{x}_j^{s-1}, \mathbf{x}_i^{s-1}, \mathbf{x}_k^{s-1})$ for the neighbors, the anchors and the imposters, respectively.

Line 6: The single-sample test (denoted by *SST*) described in Section 4.1.2.1 is performed using $\mathbf{x}^{s-1}$ and estimate the ground truth confidence scores $c^{s-1} = (c_j^{s-1}, c_i^{s-1})$ for both the neighbors and the anchors.

Line 7: $(\mathbf{z}_j, \mathbf{z}_i)$ is fed to the performance predictor to obtain the predicted confidence score $f_p(z_j, z_i)$. Then the binary cross-entropy loss denoted by L_p was computed and the respective weights θ_p were updated using SGD.

Line 8: The anchor middle-level features z_i were fed forward through the performance predictor to obtain the confidence predictions $\hat{c}_i^s$.

Line 9: The confidence score predictions $\hat{c}_i^s$ were utilized along with the high-level features $\mathbf{x}^{s-1}$ to compute the weighted-by-confidence point-to-set triplet loss in Equation (4.2). Using this loss the weights of the Feature Extractor II were updated and then the process was repeated by fetching the next batch of samples.

4.2.2 Testing

At test time, the high-level feature representations and confidence scores of the gallery $\mathcal{G} = \{(\mathbf{x}_m, l_m, \hat{c}_m) \mid m \in [1, N_\mathcal{G}]\}$ were computed, which contains only one image per subject. For a probe image set, the high-level representation and confidence score for each image

$$\boldsymbol{P} = \{(\mathbf{x}_n, \hat{c}_n) \mid n \in [1, N_{\boldsymbol{P}}]\} \tag{4.9}$$

are computed, where n is the index of the image in the set. $\boldsymbol{P}$ is a set of images belonging to the same subject and $N_\mathcal{G}, N_\mathcal{P}$ are the number of samples in the gallery and the probe, respectively. Then, for each image in the gallery, $\mathbf{x}_m$, the distance from the probe image set $\boldsymbol{P}$ was computed as

follows:

$$\mathcal{D}\left(\mathbf{x}_m, \boldsymbol{P}\right) = \frac{\sum_{n=1}^{N_{\mathcal{P}}} \hat{c}_n \mathcal{D}\left(\mathbf{x}_m, \mathbf{x}_n\right)}{\sum_{n=1}^{N_{\mathcal{P}}} \hat{c}_n}, \tag{4.10}$$

where $\mathcal{D}\left(\mathbf{x}_m, \mathbf{x}_n\right)$ corresponds to the Euclidean distance between the single image in the gallery and the n^{th} image $\mathbf{x}_n$ of the probe set. The final distance is a fusion of point-to-point distances weighted by the corresponding confidence score $\hat{c}_n$. The point-to-set distance was computed for every image in the gallery and the one with the minimum distance was selected.

4.3 Experiments

4.3.1 Datasets

The IARPA Janus Benchmark A (IJB-A) [41] and UHDB-31 [43] datasets were selected to evaluate the performance of CDN for multi-probe face recognition.

IJB-A: The IARPA Janus Benchmark A (IJB-A) [41] dataset comprises $5,397$ still images and $2,042$ videos from 500 subjects. The original protocol was designed for set-to-set face matching. To simulate a multi-probe face recognition paradigm, the $1:N$ protocol was revised. In every split, one image was randomly sampled for each subject to form the new "search-gallery". The rest of the samples were split into sets with three images from the same subject in each to form the new "search-probe". This dataset was used to assess the performance of the proposed approach "in the wild".

UHDB-31: The UHDB-31 [43] dataset comprises 77 subjects. For each subject, a still image was captured from 21 poses, under three different illumination conditions. The original protocol was designed for point-to-point face recognition. To simulate a multi-probe face recognition paradigm,

the frontal face of each subject is enrolled into the gallery. A set of 3 images from different poses were sampled and used as a probe. Details about the set sampling rules are provided in Figure 5.4 and Table 6.2. Since the size of this dataset is small, training is not performed, but instead, only testing was conducted among CDN and the state-of-the-art approaches. This dataset was used to assess the performance of proposed approach in a "controlled" environment.

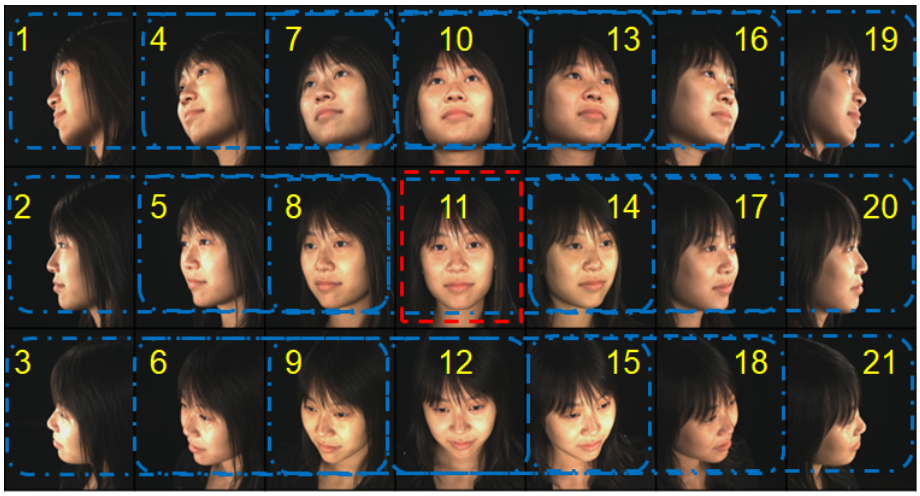

Figure 4.3: Using pose 11 as the only image in the gallery, 15 different probe sets are constructed. For example, the first set contains the poses $[1, 4, 7]$ the second $[4, 7, 10]$ and so on. In order to ensure that each pose appears the same amount of times in the probe image set, and since pose 11 dose not appear in probe image sets, some sets are constructed with just two images (e.g., $[5, 8, \text{NaN}]$).

4.3.2 Baselines

The center loss for face recognition (CLFR) of Wen et al. [79], and the quality aware etwork (QAN) of Liu et al. [47] were selected as baselines. For CLFR, the model pretrained on the CASIA WebFace database [87] was used, and fine-tuned on IJB-A. Since CLFR is designed for point-to-point matching, average fusion at a score level is performed for multi-probe face identification.

QAN was designed for set-to-set matching. However, it can be easily adapted to point-to-set matching, by applying the quality scores only to the set data. Since the pre-trained model of QAN was not available, the code provided by the authors[1] was used to train QAN from scratch on WebFace, and then fine-tune it on IJB-A. To conduct a fair comparison, CDN is trained exactly under the same protocol.

Table 4.1: Summary of rank-1 accuracy (%) results for Experiment 1.

Method	IJB-A	UHDB-31		
		I01	I03	I05
QAN [47]	71.53 ± 2.65	63.25	94.22	89.19
CLFR [79]	83.74 ± 2.72	96.42	98.41	96.38
CDN	**84.56** ± 2.78	**97.41**	**99.47**	**97.18**
p value	**0.0006**		**0.00001**	

For IJB-A, the values denote average and standard deviation rank-1 identification rate over the ten splits. In UHDB-31, testing is conducted (no training is performed) under three different illumination conditions: I01, I03, and I05 which correspond to lighting originating from the left, the central and the right side, respectively. The last row shows the performance difference test results between CDN and CLFR in 10-fold cv paired t-test on IJB-A, and 3-fold cv paired t-test in UHDB-31.

Experiment 1: The objective of this experiment was to evaluate the identification performance of CDN against state-of-the-art approaches. For the IJB-A dataset, average results over the ten splits are reported. For UHDB-31, tests were conducted under three different illuminations independently (i.e. I01, I03, and I05 which correspond to lighting originating from the left, the central, and the right side, respectively). The corresponding rank-1 accuracy (%) results are reported in Table 6.1. CDN achieved higher performance in both datasets. To verify whether the performance differences are statistically significant, a 10-fold cv paired t-test was conducted on IJB-A, and a 3-fold cv paired t-test is conducted on UHDB-31. In particular, the performance of CDN and the baseline with CLFR were compared. Corresponding $p-$values are reported in Table 3.3. At a significant level of

[1] https://github.com/sciencefans/Quality-Aware-Network

0.01, CDN achieves significant better performance The performance in both datasets. One possible reason for this is that the model used in the original QAN paper is significantly more complicated and was trained with additional commercial data. To better understand where the performance gain was originating from, additional experiments and ablation studies were conducted.

Table 4.2: Rank-1 rate for sets comprising different poses.

Set	Rank-1 Rate (%)		
	QAN [47]	CLFR [79]	**CDN**
$[1, 4, 7]$	75.95	94.81	**97.40**
$[4, 7, 10]$	86.25	**98.70**	**98.70**
$[7, 10, 13]$	90.00	**98.70**	**98.70**
$[10, 13, 16]$	91.25	**98.70**	**98.70**
$[13, 16, 19]$	83.75	96.10	**97.40**
$[2, 5, 8]$	94.94	**97.40**	**97.40**
$[5, 8, \text{NaN}]$	94.94	**97.40**	**97.40**
$[8, \text{NaN}, 14]$	95.00	**98.70**	**98.70**
$[\text{NaN}, 14, 17]$	95.00	97.40	**98.70**
$[14, 17, 20]$	95.00	97.40	**98.70**
$[3, 6, 9]$	86.25	88.31	**94.81**
$[6, 9, 12]$	91.25	**98.70**	**98.70**
$[9, 12, 15]$	91.25	**98.70**	**98.70**
$[12, 15, 18]$	91.25	**98.70**	**98.70**
$[15, 18, 21]$	85.00	93.51	**96.10**

Experiment 2: The objective of this experiment was to assess the performance improvement across sets comprising different poses. The UHDB-31 dataset was used from which sets with three (and sometimes two) different poses were formed as depicted in Figure 5.4. The obtained results from the central illumination are provided in Table 6.2. The results for the other two illuminations are

included in the supplementary material and are consistent with the central ones provided. In all 15 sets, CDN performed better (or equally) than the other methods, and demonstrated superior performance in sets that comprise large poses.

Table 4.3: Rank-1 rate (%) accuracy results when the set comprises images of the same pose but different illuminations. For example: $[(P1, I01), (P1, I03), (P1, I05)]$.

Method	P1	P4	P7	P10	P13	P16	P19
QAN	22.67	69.23	83.33	89.87	84.81	67.09	16.67
CLFR	**58.33**	**90.67**	100.00	100.00	100.00	92.11	73.33
CDN	52.78	88.00	100.00	100.00	100.00	92.11	**77.33**
	P2	P5	P8	P11	P14	P17	P20
QAN	56.41	91.03	96.20	96.20	96.20	89.87	63.29
CLFR	84.00	**96.00**	98.68	100.00	100.00	96.05	85.53
CDN	84.00	94.67	98.68	100.00	100.00	**97.37**	**86.84**
	P3	P6	P9	P12	P15	P18	P21
QAN	25.97	64.10	86.08	94.94	88.61	70.51	30.77
CLFR	**54.05**	85.33	100.00	100.00	98.68	85.33	42.67
CDN	47.30	**86.67**	100.00	100.00	98.68	**89.33**	**46.67**

Experiment 3: The objective of this experiment was to investigate how CDN performs when sets contain images of varying illumination conditions. Each set was selected such that it contained three images of the same pose but with one image per illumination. An example for pose 4 is $[(P4, I01), (P4, I03), (P4, I05)]$. Rank-1 identification rate for each pose is reported in Table 4.3 for all three methods. CDN achieved higher accuracy in $^{17}/_{21}$ cases.

Experiment 4: The objective of this experiment was to assess the impact of the size of the image set. All three approaches were evaluated when the set comprises three and six images. The obtained results are reported in Figure 4.4. Most improvements of CDN against the other two approaches are

observed in sets with large poses. The performance of the proposed approach is consistent across sets with different poses which is not the case for CLFR and QAN. For a set of six images CDN's performance is still superior to the rest of the methods but not as much as with sets of three. A reason for this is that when sets comprise six images, it's likely that more informative images will be included and a weighting scheme is less important in such cases.

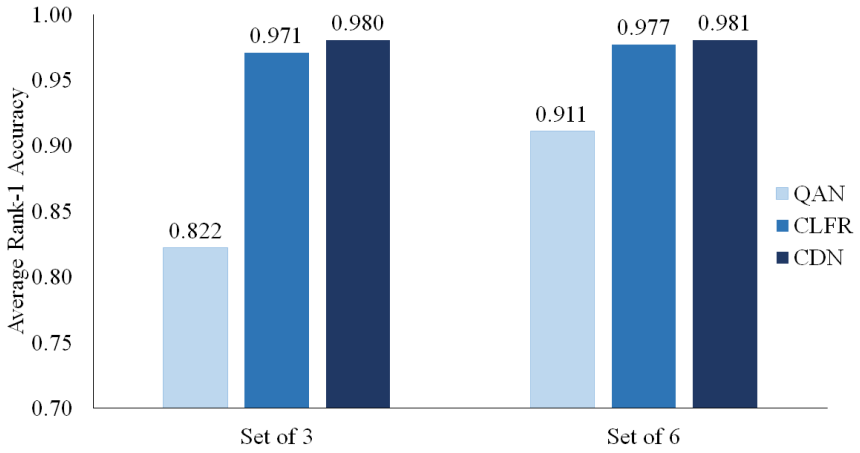

Figure 4.4: Impact of image set size on the rank-1 accuracy.

Table 4.4: Ablation studies to assess the impact of: (i) the weighted-by-confidence point-to-set triplet loss ($\mathcal{L}_{\mathcal{B}}$), and (ii) the performance prediction network (PPN).

Module			Dataset	
FEN	$\mathcal{L}_{\mathcal{B}}$	PPN	IJB-A	UHDB-31
√			83.74 ± 2.71	97.07
√		√	83.89 ± 2.50	97.75
√	√		84.06 ± 2.57	97.42
√	√	√	**84.56** ± 2.78	**98.02**

Experiment 5: The objective of this experiment was to verify the contribution of different design components proposed in CDN framework : (i) the weighted-by-confidence point-to-set triplet loss

($\mathcal{L}_{\mathcal{B}}$), and (ii) the performance prediction network (PPN). The ablation studies were conducted, and results are reported in Table 4.4. On the first row, all the proposed components are removed which results in the CLFR approach trained on the WebFace database [87]. By fine-tuning CLFR on IJB-A while maintaining the center loss, an average rank-1 accuracy of 83.74% was obtained over the 10 splits. On the second row, fine-tuning of the feature extraction network on the IJB-A database was removed. Instead, the performance prediction network which was fine-tuned with its binary cross-entropy loss was plugged. By doing so, an absolute increase in the performance of 0.15% was observed. On the third row, only the feature extraction network was fine-tuned on IJB-A with triplet loss. The rank-1 accuracy of 84.06% was obtained. One the last row, by jointly fine-tuning both modules, a relative improvement of 0.59% was obtained over FEN. Both modules contributed to the final improvements.

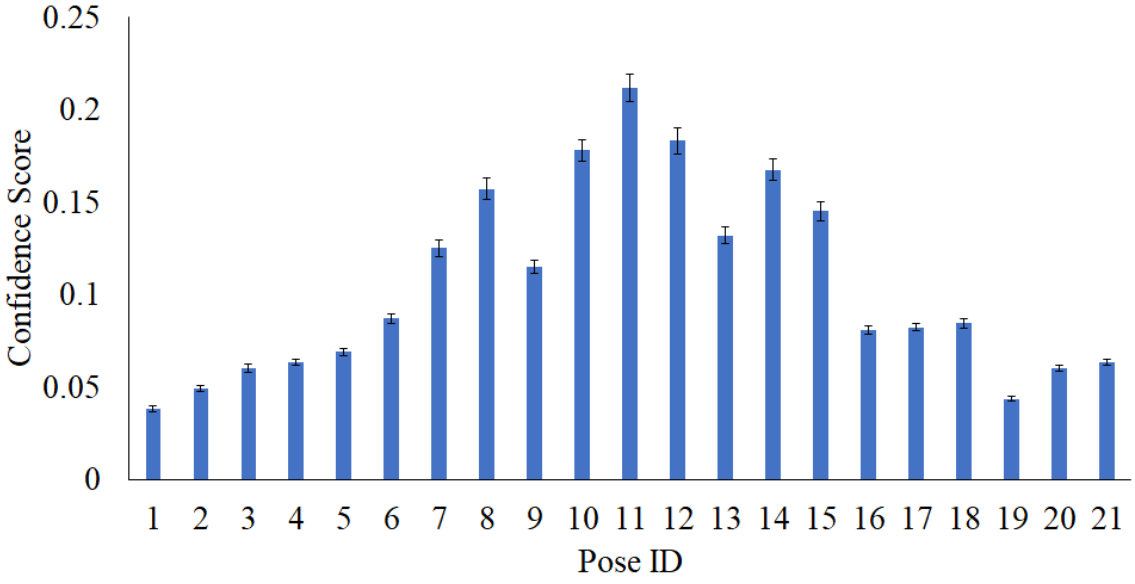

Figure 4.5: Confidence score estimates of the performance predictor under different poses when tested on the UHDB-31 database.

Experiment 6: In Figure 4.5 the confidence score predictions of PPN are reported for the UHDB-31 dataset averaged for each pose along with their standard error. The performance predictor

Table 4.5: Confidence score estimates of the performance predictor under three different illuminations when tested on the UHDB-31 database.

Illumination	Confidence Score
Left Side	0.10 ± 0.0016
Center	0.14 ± 0.0027
Right Side	0.09 ± 0.0014

provides on average higher confidence to images with near-frontal poses. The standard error for near-frontal poses (i.e. Pose IDs 10, 11, 12) was at least twice as much compared to the larger poses (i.e. Pose IDs 1, 2, 3, 19, 20, 21). The reason for this is that there are other factors (such as skin color or illumination) besides the pose that affects the identification performance. In large poses, the pose is the main reason for the performance drop and thus, the variation is smaller. Next, let's focus on the three different sources of lighting for which the results are provided in Table 4.5. The performance predictor favors center lighting on average and performs similarly in the other two illumination conditions. Note that the reason the confidence scores are low is because they are absolute unnormalized values and that CDN has not been trained on this dataset.

Experiment 7: Looking solely at aggregated pose and illumination results did not provide a full picture of what the performance predictor was learning. Towards this direction, qualitative results are proposed in Figures 4.6 and 4.7. From Figure 4.6, it can be observed that: (i) within a subject, pose and illumination influence the confidence score, (ii) the confidence score is distributed differently for different subjects. In Figure 4.7 (b,c), it is demonstrated that: (i) for the same subject CDN assigns significantly less confidence to images with occlusions or blur, and (ii) when both pose and illumination conditions are kept the same, different subjects can have $\times 3$ higher confidence than others.

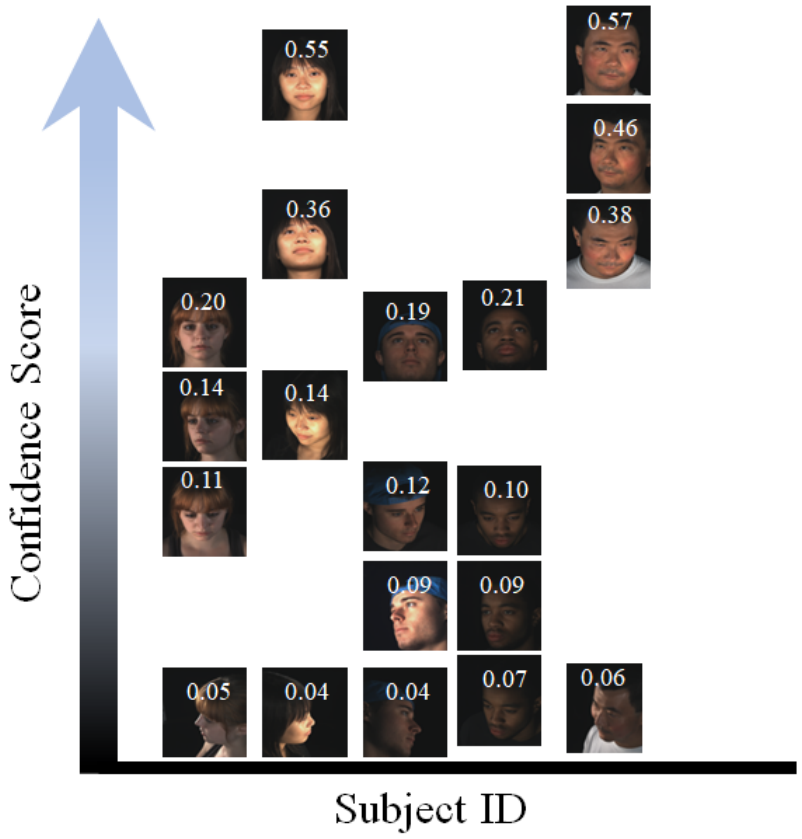

Figure 4.6: Different images of various subjects are provided with their corresponding confidence scores. Variations include skin-color, pose, illumination, and other subject-specific attributes.

Figure 4.7: Top row (a): Ranking of randomly selected images from low to high confidence from the IJB-A dataset. Bottom row (b): For the same subject CDN puts more emphasis on samples that do not suffer from occlusions or image blur. Bottom row (c): For the same pose and the same illumination (left, center, and right, respectively) different subjects demonstrate $\times 3.5$ higher confidence scores.

Chapter 5

Objective 3: Enhance Sample-based Face Recognition System for Set-based Tasks

In this chapter, the proposed confidence-driven network was extended into an add-on module which can adapt a sample-based face recognition (FR) system for set-based FR applications and enhance the performance. In particular, the confidence prediction network (CPN) is proposed. Similar to CDN, CPN follows the attention-based model and utilize CPN to generate confidence scores as attention. As a result, it is (i) free from model assumptions, and (ii) computationally inexpensive. The batch-based single-sample-test mechanism was extended to generate the global pseudo-ground-truth for the confidence scores such that the confidence scores can be learned (i) independently without the access to the template of a sample-based FR system, and (ii) without set-based restriction in the training batch. Compared with CDN, the proposed CPN has the following differences: (i) the training of feature representations and the confidence score are completely independent which simplifies the training process a lot, (ii) CPN can work with different face recognition systems on both feature level and score level, (iii) comprehensive experiments are conducted for both

point-to-set matching and set-to-set matching, and (iv) different attention mechanisms are compared under the same backbone network with the same amount of training data.

5.1 Method

The proposed confidence prediction network consists of two parts, a sample-based FR system, and a performance prediction network. The sample-based FR system could be any well-trained FR system. Given two face images as input $(\boldsymbol{I}_t, \boldsymbol{I}_i)$, the FR system will output a similarity score $s_{ti} = \psi_s(\boldsymbol{I}_t, \boldsymbol{I}_i)$. The performance prediction network is learned to pair with the sample-based FR system. Given one face image, $\boldsymbol{I}_t$, as input, the performance predictor will output a confidence score c_t which measures the confidence level of the input image under the corresponding sample-based FR system. In this section, I explained in details how to train a confidence prediction network, and how it can be used in set-based matching.

5.1.1 Training

In the training process, it is necessary to have access to (i) a sample-based FR system $\psi_s(\boldsymbol{I}_i, \boldsymbol{I}_t) = s_{it}$ which can return the similarity score s_{it} between two images (I_i and I_t), and (ii) a set of training images $\{(\boldsymbol{I}_t, l_t) \mid t \in [1, T], l_t \in [1, N_C]\}$, where l_t denotes identity label and N_C is the total number of classes. The objective is to learn the mapping $\phi()$ from a input image $\boldsymbol{I}_t$ to its corresponding confidence score c_t. In this work, a two-step approach is proposed, (i) generate pseudo-ground-truth for target confidence score, and (ii) train a regressor to regress from the input image $\boldsymbol{I}_t$ to target confidence score $\hat{c}_t$.

Step 1: Pseudo ground truth. A global single-sample-test mechanism is proposed to simulate

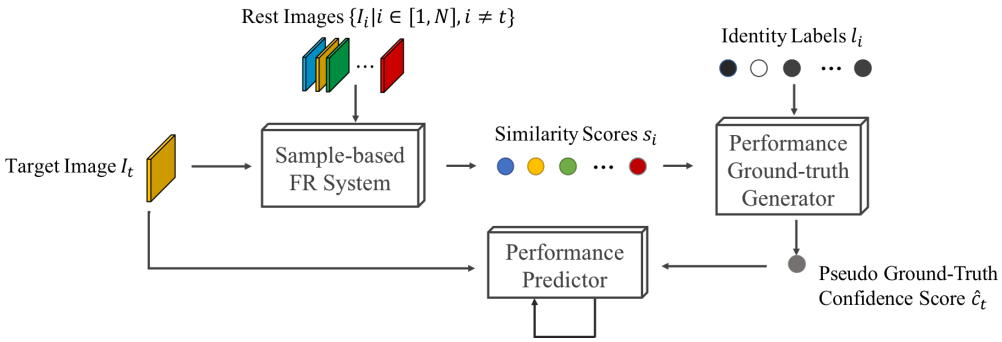

Figure 5.1: Illustration of the training strategy of CPN.

the matching process and generate global pseudo-ground-truth confidence score $\hat{c}_t$ for each training image $\boldsymbol{I}_t$. The steps to compute $\hat{c}_t$ for input image $\boldsymbol{I}_t$ is quite straightforward.

(i) The similarity score s_{ti} of between $\boldsymbol{I}_t$ and the rest of images $\boldsymbol{I}_i, i \neq t$ in training dataset is computed under the samplbe-based FR system via

$$s_{it} = \psi_s \left(\boldsymbol{I}_t, \boldsymbol{I}_i \right). \tag{5.1}$$

(ii) The ground truth for the pair-wised similarity score between $\boldsymbol{I}_t$ and the rest of training images $\boldsymbol{I}_i, i \neq t$ is computed via

$$\hat{s}_{it} = \begin{cases} 0 & \text{if } l_i = l_t \\ 1 & \text{if } l_i \neq l_t \end{cases}. \tag{5.2}$$

(iii) The ROC curve using predicted scores, s_{it}, and the ground-truth similarity $\hat{s}_{it}$ are computed. The area under the curve is used as the pseudo ground-truth confidence score for $\boldsymbol{I}_t$.

The proposed approach is based on the assumption that the confidence level of a sample under a specific sample-based FR system can be measured by its performance when it is compared with other samples. The higher the performance, the more confident the system is about this sample. The

AUC of the roc curve was selected as the similarity measurement. Because in the proposed single-sample-test, the pair-wise comparisons were highly imbalanced. There were far more dissimilar pairs than similar pairs. The only restriction of this approach on the training data was that there were at least two samples for each class. Compared with the single-sample-test used in CDN, the global single sample test shared two major differences. First, the test was conducted globally across the whole training dataset. With more samples included, more robust performance measurements were expected. Second, the AUC was used as the performance measurement instead of accuracy. The AUC was more robust under highly imbalanced data compared with the average accuracy.

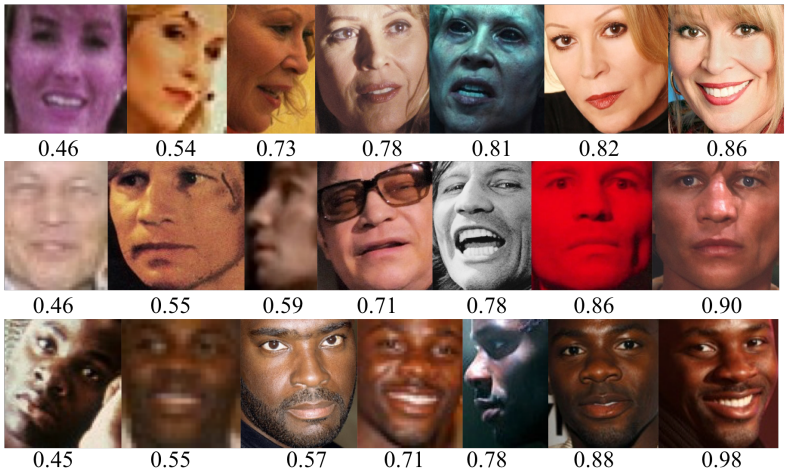

Figure 5.2: Examples of the pseudo-ground-truth confidence scores. Each row is a face image set containing seven images from the same person. As observed, images in the set exhibit large variations (e.g., resolution, poses, makeups, illuminations, etc). The number under each image is its corresponding pseudo-ground-truth confidence score generated for ArcFace [14] according to the proposed method described in Sec. 5.1.1.

Step 2: Prediction. Now for each training image $\boldsymbol{I}_t$ there is a target pseudo-ground-truth-confidence-score $\hat{c}_t$. The task is to learn a regression $\phi()$ from the training image $\boldsymbol{I}_t$ to the target confidence score $\hat{c}_t$. In this work, the standard ResNet18 was used to model $\phi()$. Because $\hat{c}_t$ can be considered as the possibility that sample $\boldsymbol{I}_t$ will return a correct prediction, simple cross-entropy

loss was used for optimization

$$\mathcal{L} = -\hat{c}_t \log(\phi(\boldsymbol{I}_t)) - (1 - \hat{c}_t) \log(1 - \phi(\boldsymbol{I}_t)) \tag{5.3}$$

A summary of the training training process is provided in Algorithm 5.1.

Algorithm 5.1: CPN: Training

Input : Training Data $\{(\boldsymbol{I}_t, l_t)|l \in [1, T]\}$ and sample based FR system $\psi_s()$.
Output: Parameters θ for performance prediction network $\phi()$.

/* Generate Pseudo Ground Truth $\hat{c}t$ */
for $t \leftarrow 1$ *to* T **do**
  for $i \leftarrow 1$ *to* T *and* $i \neq t$ **do**
    $s_{it} \leftarrow \psi_s(\boldsymbol{I}_i, \boldsymbol{I}_t)$;
    if $l_i = l_t$ **then**
      $\hat{s}_{it} \leftarrow 1$
    else
      $\hat{s}_{it} \leftarrow 0$
    end
  end
  $\hat{c}_t \leftarrow AUC(\{(s_{it}, \hat{s}_{it})|i \in [1, T], i \neq t\})$;
end
/* Train the predictor $\phi()$ */
Initialization: step $t = 0$, θ_t ;
while *(validation loss decreases)* **do**
  $t \leftarrow t + 1$;
  $c_t \leftarrow \phi(\mathbf{I}_t, \theta_{t-1})$; // Feed-forward of performance predictor
  $\theta_t = SGD(\mathcal{L}(\hat{c}_t, c_t), \theta_{t-1})$; // $\mathcal{L}$ is defined in Equation 5.3
end

5.1.2 Set-based Matching

The objective of this subsection is to discuss how to use the learned CPN to enhance the performance of set-based matching. Given two image sets $\mathcal{S}_1 = \{\boldsymbol{I}_i\}_{i=1}^{N_1}$, $\mathcal{S}_2 = \{\boldsymbol{I}_j\}_{j=1}^{N_2}$, and a sample-based FR system, the task is to measure the similarity s_{12} between these two image sets.

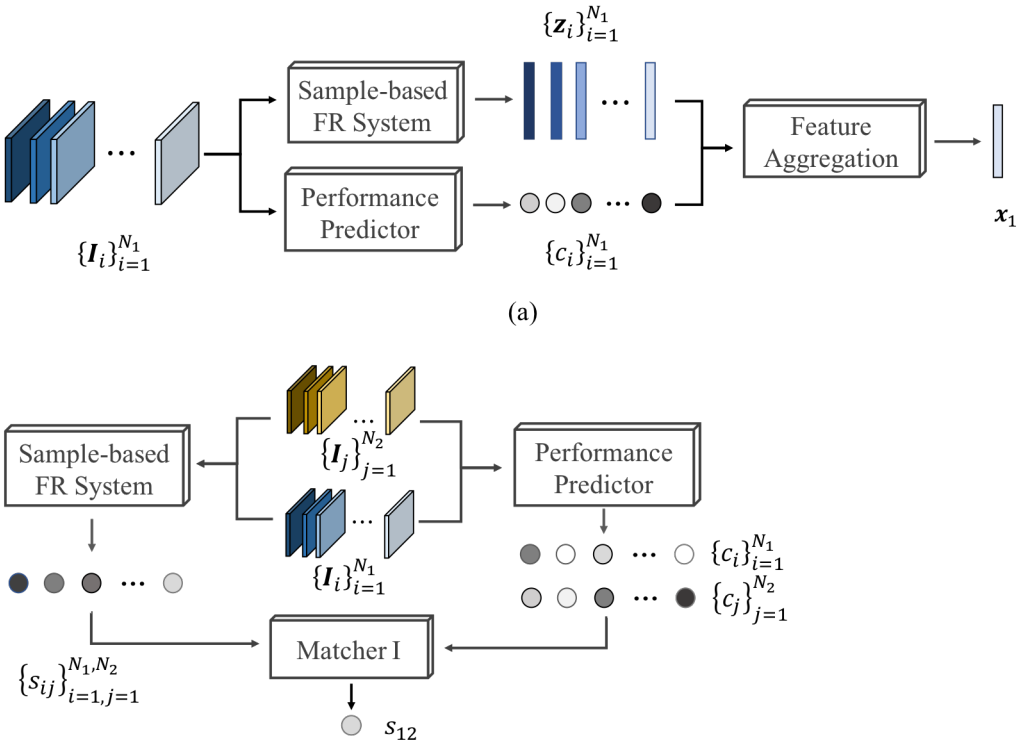

Figure 5.3: Illustration of set-based matching with CPN. (a): Feature-level aggregation when templates from the sample-based FR system is available. (b): Score-level aggregation when only similarity scores are available for the sample-based FR system.

In this work, two situations were considered, (i) access to the template generated by the sample-based FR system $\boldsymbol{z}_i = \psi_f(\boldsymbol{I}_i)$, and (ii) access to the similarity scores. As illustrated in Figure 5.3, the feature-level aggregation and score-level aggregation can be applied, respectively. In the first situation, the template $\boldsymbol{z}_i$ from the sample-based FR system, and the confidence score c_i from CPN were available for each sample in set. Then the set-based template can be aggregate via,

$$\boldsymbol{x}_1 = \frac{\sum_i c_i \boldsymbol{z}_i}{\sum_i c_i}. \tag{5.4}$$

Algorithm 5.2: CPN: Feature-level Aggregation

Input : Testing Image Set: $\mathcal{S}_1 = \{\boldsymbol{I}_i\}_{i=1}^{N_1}$ and $\mathcal{S}_2 = \{\boldsymbol{I}_j\}_{j=1}^{N_2}$.
Output: Similarity Score s_{12} between $\mathcal{S}_1$ and $\mathcal{S}_2$.

/* Feature Aggregation for $\mathcal{S}_1$ */
for $i \leftarrow 1$ *to* N_1 **do**
  $\boldsymbol{z}_i = \psi_f(\boldsymbol{I}_i)$;
  $c_i = \phi(\boldsymbol{I}_i)$;
end
$\boldsymbol{x}_1 = \frac{\sum_i c_i \boldsymbol{z}_i}{\sum_i c_i}$;
for $j \leftarrow 1$ *to* N_2 **do**
  $\boldsymbol{z}_j = \psi_f(\boldsymbol{I}_j)$;
  $c_j = \phi(\boldsymbol{I}_j)$;
end
$\boldsymbol{x}_2 = \frac{\sum_j c_i \boldsymbol{z}_j}{\sum_j c_j}$;
/* Similarity Computation */
$s_{12} = \frac{\boldsymbol{x}_1 \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \|\boldsymbol{x}_2\|}$

Same steps were applied to obtain $\boldsymbol{x}_2$ for $\mathcal{S}_2$. Then the similarity between $\mathcal{S}_1$ and $\mathcal{S}_2$ was computed via

$$s_{12} = \frac{\boldsymbol{x}_1 \boldsymbol{x}_2}{\| \boldsymbol{x}_1 \| \| \boldsymbol{x}_2 \|}. \tag{5.5}$$

A summary of the feature-level aggregation process is provided in Algorithm 5.2.

In the second situation, the following items were computed: (i) the pairwise similarity scores between the two image sets (i.e. $\{s_{ij} | i \in [1, N_1], j \in [1, N_2]\}$), and (ii) the confidence scores for each sample in two sets (i.e. $\{c_i | i \in [1, N_1]\}$ and $\{c_j | j \in [1, N_2]\}$). The final confidence score is computed via

$$\frac{\sum_j \sum_i c_i c_j s_{ij}}{\sum_j \sum_i c_i c_j}. \tag{5.6}$$

A summary of the feature-level aggregation process is provided in Algorithm 5.3.

Algorithm 5.3: CPN: Score-level Aggregation

Input : Testing Image Set: $\mathcal{S}_1 = \{\boldsymbol{I}_i\}_{i=1}^{N_1}$ and $\mathcal{S}_2 = \{\boldsymbol{I}_j\}_{j=1}^{N_2}$,
Sample-based FR system $\psi_s()$.
Output: Similarity Score s_{12} between $\mathcal{S}_1$ and $\mathcal{S}_2$.

/* Pair-wise Similarity Scores */
for $i \leftarrow 1$ *to* N_1 **do**
  for $j \leftarrow 1$ *to* N_2 **do**
    $s_{ij} = \psi_s(\boldsymbol{I}_i, \boldsymbol{I}_j)$;
  end
end
/* Confidence Score Computation */
for $i \leftarrow 1$ *to* N_1 **do**
  $c_i = \phi(\boldsymbol{I}_i)$;
end
for $j \leftarrow 1$ *to* N_2 **do**
  $c_j = \phi(\boldsymbol{I}_j)$;
end
/* Score-level Aggregation */
$s_{12} = \frac{\sum_j \sum_i c_i c_j s_{ij}}{\sum_j \sum_i c_i c_j}$

5.2 Experiments

5.2.1 Datasets

In this chapter, all the algorithms were trained on the IMDb dataset [69], and tested in the IARPA Janus Benchmark-C (IJB-C) [54] for set-to-set matching, and UHDB-31[43] dataset for point-to-set matching.

IMDb-face: The IMDb-face dataset is a new large-scale noise-controlled dataset, which comprises about 1.7 million faces from 59 k identities. The face images were collected from the IMDb website and manually cleaned. The IMDb-face dataset was selected for training because it is a noise-controlled dataset. Wang et al. [69] claimed that label noise in training has a negative impact on

the face recognition performance. In the experiments, 20% of the subject were randomly sampled to form the validation split, the rest of the subjects were used for training.

IJB-C: The IARPA Janus Benchmark C (IJB-C) [41] dataset was selected to evaluate the performance of set-to-set matching in the uncontrolled environment. It comprised $31,334$ still images and $11,779$ videos from $3,531$ subjects. The IJB-C dataset was selected for two reasons. First, it is the most challenging dataset for set-based face recognition. Second, except for the identity information, six groups of meta-information (i.e. face size, facial hair, age, indoor/outdoor, skin tone, gender, and pose) are also provided. The meta-information was employed to further understand the learned confidence scores and performance bias on the current FR system. The original protocols of IJB-C were designed for set-to-set matching for verification, close-set identification, and open-set identification.

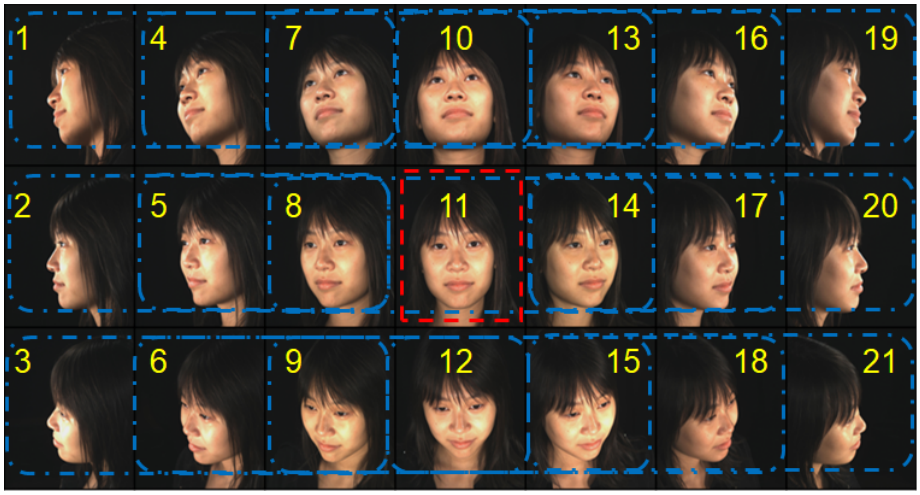

Figure 5.4: Using pose 11 as the only image in the gallery, 15 different probe sets are constructed. For example, the first set contains the poses $[1, 4, 7]$, the second $[4, 7, 10]$, and so on. In order to ensure that each pose appears the same amount of times in the probe image set, and since pose 11 dose not appear in any probe image sets, some sets were constructed with just two images (e.g., $[5, 8, \text{NaN}]$).

UHDB-31: The UHDB-31 [43] dataset comprises of 77 subjects. For each subject, a still image

was captured from 21 poses, under three different illumination conditions. This dataset was used to assess the performance of the point-to-set matching in a "controlled" environment.

5.2.2 Baselines

The additive angular margin loss for deep face recognition (ArcFace) proposed by Deng et al. [14] was selected as the backbone network for the sample-based RF system, because it achieves the state-of-the-art performance in sample-based face-recognition tasks. Two other set-based approaches, the multicolumn networks (MN) proposed by Xie et al. [83], and the neural aggregation network (NAN) proposed by Yang et al. [85] were selected. These two algorithms were selected because: (i) they achieved the state-of-the-art performance in set-based face recognition tasks, and (ii) similar to CPN, they were also employed to learn the attention distribution and conduct attention-based set aggregation. For Arcface, the pretrained model provided by the authors was employed. For MN and NAN, the networks are implemented using Pytorch according to the original papers. To conduct a fair comparison, the same backbone network (i.e. ArcFace), training data (i.e. IMDB) were used for CPN, MN, and NAN. To better align with the ArcFace template, MN and NAN were trained using the additive angular margin loss function proposed in ArcFace with the learned attention scores as weights. With these baselines, the objective is to figure out: (i) whether the confidence score learned from CPN can help improve the performance of ArcFace, and (ii) how CPN performs compared with other set-based approaches.

5.2.3 Experimental Results

Set-to-Set Matching: The objective of this experiments was to assess the performance of CPN on the task of set-to-set matching. The performance of Arcface, NAN, MN, and CPN were tested.

Table 5.1: Performance evaluation on the IJB-C dataset: 1:1 verification TAR(%).

Method	1:1 Verification TPR (%)					
	FPR$= 1E-6$	FPR $= 1E-5$	FPR $= 1E-4$	FPR $= 1E-3$	FPR $= 1E-2$	FPR $= 1E-1$
ArcFace	79.80	85.75	90.23	93.48	95.94	98.10
MN	64.92	74.85	83.02	89.24	94.03	97.46
NAN	78.21	84.54	88.97	92.41	95.31	97.71
CPN	**80.04**	**86.56**	**90.79**	**93.70**	**96.03**	**98.16**
Delong Test on AUCs (CPN vs. ArcFace): $p = 0.41 > 0.005$						

Table 5.2: Performance evaluation on the IJB-C dataset: 1:N identification

Method	Open-Set Identification TPIR(%)			Close-Set Identification ACC(%)		
	FPIR $= 1E-5$	FPIR$= 1E-3$	FPIR $= 1E-1$	Rank 1	Rank 5	Rank 10
ArcFace	85.77	93.51	98.10	91.02	93.55	94.45
NAN	84.51	92.44	97.72	90.19	92.93	93.75
MN	75.74	89.48	97.56	88.79	92.01	93.13
CPN	**86.64**	**93.73**	**98.15**	**91.46**	**93.85**	**94.70**

Delong Test on ROC for open-set identification (CPN vs. ArcFace): $p = 0.58 > 0.005$
Wilcoxon signed-rank test on full-rank scores for close-set identification (CPN vs. ArcFace): $p = 1.37E-10 < 0.05$

In the $1:1$ verification task, the true positive rate (TPR) at different false positive rates (FPR) are reported. A summary of results is presented in Table 5.1. In the $1:N$ identification task, the rank-n identification rate is reported for close-set identification. The true positive identification rates (TPIR) with different false positive identification rate (FPIR) are reported for open-set identification. Corresponding results are summarized in Table 6.1. The proposed CPN achieved higher performance over the baseline algorithms in all three tasks. To test whether the improvement is statistically significant, DeLong test was conducted to compare the AUCs for verification and open-set identification, and Wilcoxon signed-rank test for the close-set identification. The test results indicated that the AUCs of CPN and ArcFace were comparable for the verification and open-set identification. CPN achieved significantly better ranking performance in the close-set identification. The DeLong test focused on the AUC, because the AUC of ArcFace is 0.9916 which is already very high. It is difficult to get significant improvements. However, CPN obtained at least 0.05% improvement on the TPR and TPIR over all different FPRs. Considering the total number of comparisons (i.e. $15,658,489$), CPN gave correct predictions on around 7800 more cases compared with ArcFace.

Point-to-Set Matching: The objective of this experiment was to assess the performance of CPN on the point-to-set matching. The UHDB-31 dataset was selected for testing. The original protocol was designed for point-to-point face recognition. To simulate a point-to-set face recognition paradigm, the frontal face of each subject was enrolled into the gallery. A set of three images from different poses were sampled and used as a probe. Details about the set sampling rules are provided in Figure 5.4. Rank-1 accuracy is reported in Table 5.1. In all 15 sets, CDN performed better (or equally) than the other methods. Superior performance was obtained in sets that contain poses with $+30^{\circ}$ in pitch.

Table 5.3: Rank-1 rate for point-to-set matching

Set	Rank-1 Rate (%)			
	ArcFace	NAN	MN	CPN
$[1, 4, 7]$	92.64	91.77	90.91	**97.40**
$[4, 7, 10]$	98.27	97.84	98.27	**98.70**
$[7, 10, 13]$	98.27	98.27	98.27	**98.70**
$[10, 13, 16]$	98.27	98.27	98.27	**98.70**
$[13, 16, 19]$	96.54	96.54	96.54	**97.40**
$[2, 5, 8]$	97.84	97.84	97.84	97.84
$[5, 8, \text{NaN}]$	97.84	97.84	97.84	97.84
$[8, \text{NaN}, 14]$	98.27	98.27	98.27	98.27
$[\text{NaN}, 14, 17]$	97.40	97.40	97.40	97.40
$[14, 17, 20]$	97.40	97.40	97.40	97.40
$[3, 6, 9]$	96.54	94.37	96.10	96.54
$[6, 9, 12]$	98.70	98.70	98.70	98.70
$[9, 12, 15]$	98.70	98.70	98.70	98.70
$[12, 15, 18]$	98.70	98.70	98.70	98.70
$[15, 18, 21]$	97.84	97.40	97.40	97.84

Table 5.4: $p-$value of one-way ANOVA test on IJBC.

Factor	Indoor/Outdoor	Gender	skin Color
$p-$value	$2.77\ E-5$	$9.66\ E-6$	$1.43\ E-29$
Factor	Facial Hair	Age	
$p-$value	$4.57\ E-15$	0.1863	

Covariates' Effects Analysis In this experiments, the correlations between the learned confidence scores and different covariates were analyzed. The covariates are the factors that have an impact on face verification performance [48]. In particular, seven covariates (i.e. indoor/outdoor gender, skin color, facial hair, age, pose, and face size) are provided for each image in the IJB-C dataset as metadata.

The gender, indoor/outdoor, skin color, facial hair, and age were categorized. For these categorized factors, one-way analysis of variance (one-way ANOVA) was conducted to each factor independently. The objective was to test whether the learned confidence scores for different groups were significantly different. The corresponding $p-$values are reported in Table 5.4. As observed, except for the age, the learned confidence score showed statistically significant difference among different groups for the selected factors. To better analyze the preference of the confidence score, the mean and standard error of the learned confidence score with different groups are depicted in Figure 5.5. As it can be observed, the CPN tended to assign higher confidence scores to outdoor images. This observation seems to be opposite from the well-known assumption that indoor face recognition achieves better performance [48]. One of the possible reason could be the data distribution of indoor and outdoor images in the training dataset. Since a global single sample test was performed to estimate the confidence score, the global distribution of the two groups in the training data had an impact on the learned performance.

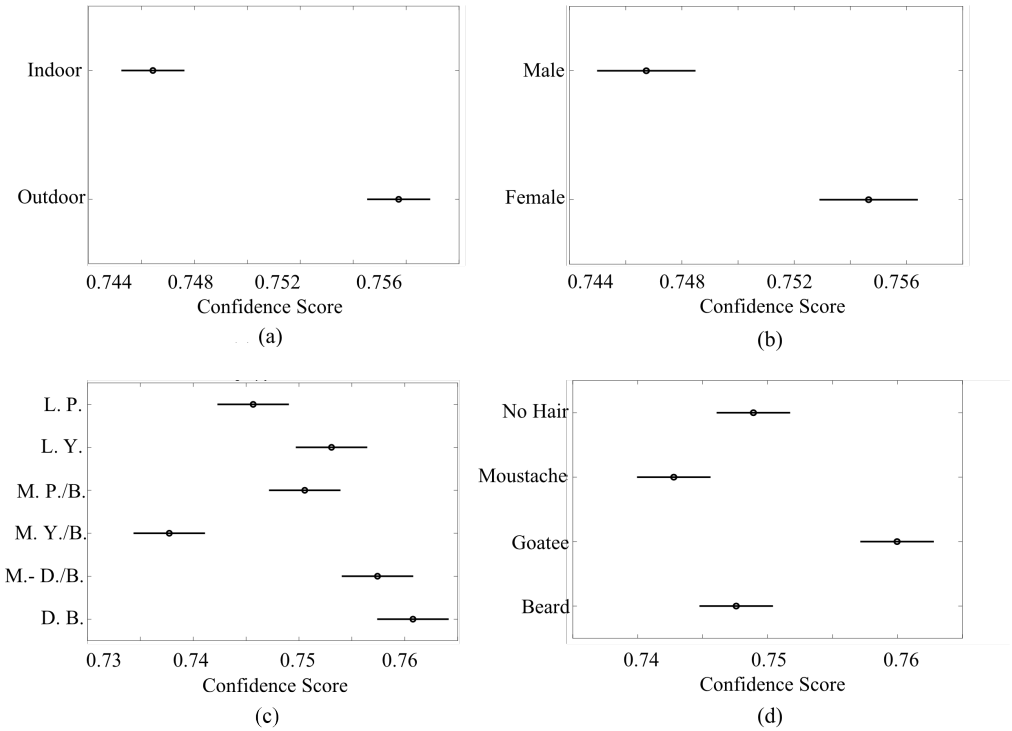

Figure 5.5: Depict of the mean and standard error of confidence scores leaned for different groups. (a): Photo environments. (b): Genders. (c): Skin colors. (d): Facial hairs. In particular, the skin color is divided into six groups: light pink (L. P.), light yellow (L. Y.), medium pink/yellow (M. P./Y.), medium yellow/brown (M. Y./B.), medium-dark brown (M.-D./B), and dark brown (D. B.). The dot is the mean value of the confidence score for each group, and short line indicates the standard error of the confidence score. If two short lines have overlaps on the x axis, the confidence score of the two corresponding groups were not significantly different.

For gender, CPN seemed to assign higher confidence scores for females compared with males. The observation aligns with the evaluation of Beveridge et al. [6, 4, 5] that females are easier to be recognized. However, studies on the impact of gender on face recognition have led to different conclusions [48]. Skin color was divided into six groups: light pink (L. P.), light yellow (L. Y.), medium pink/yellow (M. P./Y.), medium yellow/brown (M. Y./B.), medium-dark brown (M.-D./B), and dark brown (D. B.). From the results, CPN tended to assign higher confidence scores to subjects with M.-D./B and D. B. skin tone, and lower confidence scores to subjects with M. Y./B. skin tone. There are several studies analyzing the biases of the face recognition system on skin colors [1, 21, 48]. However, most of the large-scale training dataset have different populations for different skin colors. This could be the key reasons for the bias of the face recognition systems. Facial hair is grouped into no facial hair, mustache, goatee, and beard. People with goatees are more likely to be assigned a higher confidence score. People with mustaches tend to be assigned a lower confidence score.

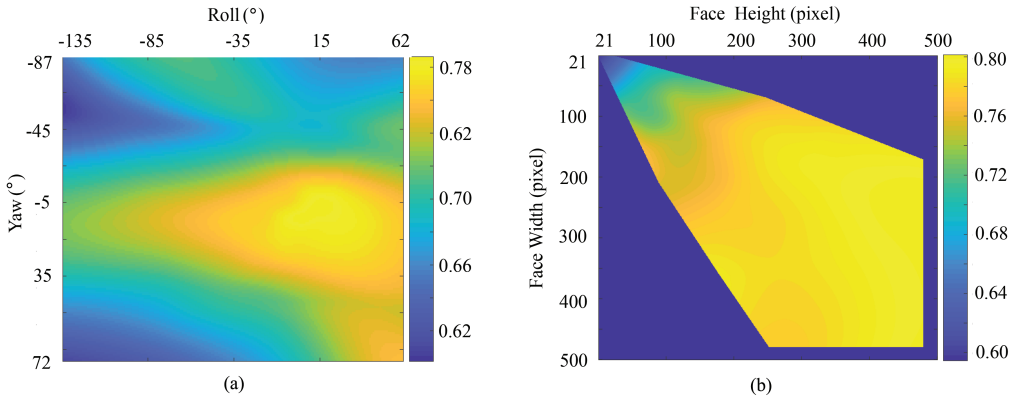

Figure 5.6: Heat maps of the average confidence score distributions regarding poses and face size. (a): Poses. (b): Face sizes.

Pose (i.e. yew and roll) and face size (i.e. face width and height) are sampled from factors with continuous values. The distribution of the mean confidence score on different poses and face sizes are visualized in Figure 5.6 using heat maps. For pose, the heat map is estimated using the local

linear regression. The yaw and roll were bounded by the minimum and maximum degrees that appeared in the IJB-C dataset. The distribution of the confidence score was symmetric regarding yaw at $0°$. The larger the degrees of yaw, the lower the confidence scores assigned. However, the confidence scores don't show the same property on the roll. The highest confidence score was around $15°$. One of the possible reason could be the pose distribution in the training set. In IMDb, the yaw angle was distributed symmetrically regarding $0°$. The roll distribution was not accessible. The face size denotes the face size in the original image. In the preprocessing steps, the face areas were cropped, and aligned to 114×114 pixel. As a result, the larger the face in the original image, the higher the resolution after alignment. The heat map of average confidence score with face size was computed in a similar way to the pose one. In particular, faces with a width or height larger than 500 pixels were removed. Because the samples were sparsely distributed beyond 500 pixels. The deep blue area close to the boundaries was unreachable face weights and height ratios. Thus, the confidence scores were not estimated. Within 500 pixels, the larger the face, the higher the confidence scores.

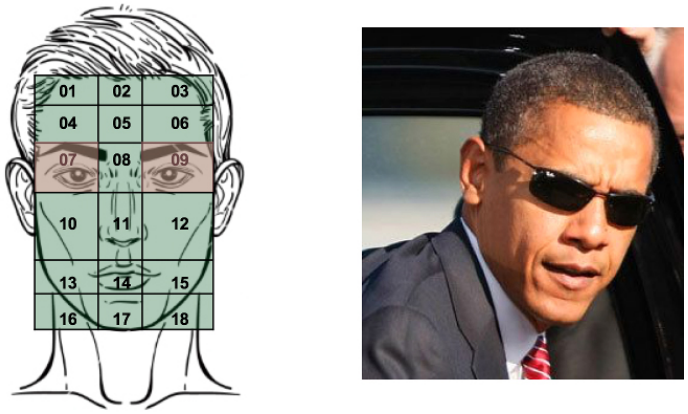

Figure 5.7: The patch splits for occlusion in IJB-C dataset. Left: the face area is split into 18 patches. Right: In the example image, eyes are occluded. The corresponding patches 07 and 09 are marked with 1 (pink), the rest of the patches are marked with 0 (green).

To represent the occlusion information, face areas were split into 18 patches as illustrated in Figure

5.7. The occlusion information was represented using an 18 dimensional binary vector. If one patch was occluded, the corresponding bit was marked as 1. To analyze if the occlusion on each patch had a statistical significant impact on the distribution of the confidence score, one-way ANOVA was conducted on to each patch independently. The corresponding $p-$ values were collected in Figure 5.8. Occlusions on patches $\{01, 02, 03, 04, 05, 08, 11, 12, 13, 17\}$ had a significant impact on the distribution of the confidence scores.

1.24E-16	3.29E-02	4.20E-36
9.95E-06	1.12E-04	1.78E-01
3.28E-01	1.33E-03	2.60E-02
8.09E-01	2.99E-05	3.83E-08
1.07E-14	4.33E-80	4.09E-02
1.67E-03	3.48E-07	9.83E-01

Figure 5.8: $p-$value of the one-way ANOVA test on patch occlusion. Left to right and up to down corresponds to patch 01 to 18 in Figure 5.7. Cells filled in grey indicate batches where occlusions did not have significant impact on the distribution of the confidence scores.

Table 5.5: Correlation analysis between confidence scores and similarity scores

Confidence Scores	Sample 1	Sample 2
Correlation Coefficients	0.0349	0.0370
$p-$value	$9.51\ E-44$	$3.69\ E-39$

Correlations with Similarity Scores: To understand the correlations between the learned confidence scores and the original similarity scores from ArcFace, a correlation analysis was conducted. The correlation coefficients and corresponding $p-$values are summarized in Table 5.5. Since a

comparison corresponds to one similarity score and two confidence scores, the correlation analysis was conducted on each of them independently. The correlation coefficients were small, which indicates that the correlation between the learned confidence scores and the similarity scores is very weak. The value of similarity scores cannot represent the confidence level of a specific prediction, and cannot be used to replace the confidence scores.

Figure 5.9: Ranking face images of the same subject according to the confidence score. Each row contains ten images randomly sampled from the same subject. From left to right, the corresponding confidence score gets higher and higher. As expected, images with high confidence scores had higher resolution, more frontal poses, and without any occlusions.

Visualization: In Figure 5.9, the within subject images are visualized and ranked by the learned confidence score. Ten images were randomly sampled from the same subject in IJB-C, and then ranked according to the learned confidence score.

Chapter 6

Objective 4: Binary Templates for Face Image Sets

In this chapter, the objective was to generate fixed length binary templates for face image sets. In the coding theory, the variable-length code is typically used for lossless data compression. To compare the similarity of variable length codes, the codes were first uncompressed, and then the similarity was computed in the original feature space. This work did not employ variable length codes. The proposed binary templates were compared directly in the binary feature space. In particular, the attention-based recursive-binary embedding (ARBE) algorithm was proposed to extract binary templates for image sets. Specifically, the network contains two parts, (i) attention-based feature learning, and (ii) recursive binary coding, as illustrated in Figure 6.1. In the first part, a real-valued feature representation was learned for each sample with a corresponding attention score. The attention score described the contribution of the corresponding sample representations. In the second part, each bit was learned recursively. The output of the previous bit was used as meta input when learning the current bit. The results from different samples were integrated at each bit.

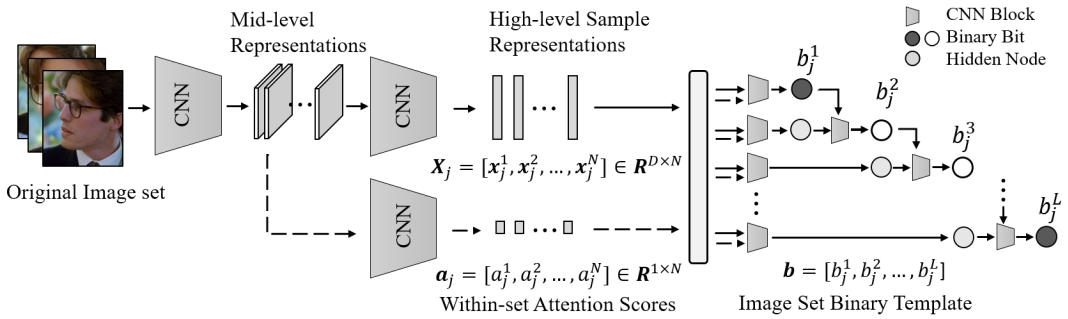

Figure 6.1: Illustration of the network architecture. The first part of the network generated a real-valued feature representation and a within-set attention score for each sample in an image set. The second part embedded the real-valued features into a compact binary template in a recursive manner.

The proposed ARBE increased recognition performance compared to the sequential code, while the number of projections was still restricted to a linear relation to code length. Learning from the recent advances in face recognition and image set classification, the angular-based similarity [46] and the image set attention schemes [47, 85] were also adapted in the proposed framework. The primary contribution is a new binary embedding framework for a face image set with the following advantages: (i) ABRE increased the recognition power while maintaining a linear model complexity, and (ii) ABRE was designed under a standard neural network architecture so that it was easily integrated into different network designs.

6.1 Methods

6.1.1 Attention-based Feature Extraction

Given an image set $\mathbf{I}_j = \left\{\mathbf{I}_j^i | i \in [1, N]\right\}$ with N images, the objective of is to extract a high-level feature representation $\mathbf{x}_j^i$ for each single image $\mathbf{I}_j^i$ with a corresponding attention score a_j^i. Similar with the idea described in [47, 85], the first CNN block $\varphi_r()$ extracts mid-level feature representations. The second CNN block $\varphi_x()$ on the top branch extracts high-level representations. The third CNN block on the bottom branch $\varphi_a()$ generates the attention scores. The attention score a_j^i indicates the contribution of its corresponding image within the set. However, the set pooling is not used to integrate the sample representations into a single real-valued template. Samples are integrated at the loss level using the attention scores. This part of the network is pretrained using a weighted angular-softmax loss [46]. The loss on a training batch $\{(\mathbf{X}_j, y_j) | j \in [1, K]\}$ is defined as

$$\begin{aligned} \mathcal{L}_f &= -\frac{1}{K}\sum_j \log\left(\frac{e^{\phi(\mathbf{X}_j, \mathbf{a}_j, \mathbf{w}_{y_j})}}{e^{\phi(\mathbf{X}_j, \mathbf{a}_j, \mathbf{w}_{y_j})} + e^{\sum_{c \neq y_j}\phi(\mathbf{X}_j, \mathbf{a}_j, \mathbf{w}_c)}}\right) \\ \phi(\mathbf{X}_j, \mathbf{a}_j, \mathbf{w}_c) &= \sum_{i=1}^{N} a^i \|\mathbf{x}_j^i\| \cos(k\theta_{i,c}), \end{aligned} \tag{6.1}$$

which can be viewed as a soft-max loss on top of an weighted angular linear layer $\phi(\cdot)$. The angular linear layer is parameterized using $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, ..., \mathbf{w}_C]^T$, where C is the total number of classes in training data, and $\|\mathbf{w}_c\| = 1$. In particular, $\theta_{j,c}$ is the angle between the feature representation $\mathbf{x}_j^i$ with the projection vector from the c^{th} classes $\mathbf{w}_c$, and k is a scaling integer. The cosine similarity $\cos(k\theta_{i,c})$ can be easily represented using expressions only containing $\mathbf{X}_j$ and $\mathbf{w}_c$ by simply using the multi-angle formula. In Equation (6.1.1), the losses from different samples $\mathbf{x}_j^i$ within set $\mathbf{X}_j$ are simply integrated on the loss level using the attention scores a_j^i with the restriction that $a_j^i \in [0, 1]$ and $\sum_{i=1}^{N} a_j^i = 1$. Ground-truth for the attention scores is not provided,

so φ_a is optimized jointly with φ_r and φ_x by minimizing Equation (6.1.1) using only the identity information in the pre-training step.

6.1.2 Recursive Binary Encoding

After the attention-based feature extraction, a real-valued feature matrix $\mathbf{X}_j = \left[\mathbf{x}_j^1, \mathbf{x}_j^2, ..., \mathbf{x}_j^N\right]$, and a corresponding attention vector $\mathbf{a}_j = \left[a_j^1, a_j^2, ..., a_j^N\right]$ are obtained. The objective is to generate a single binary template $\mathbf{b}_j = \left[b_j^1, b_j^2, ..., b_j^L\right]$ from the high-level feature representation $\mathbf{X}_j$, where L is the code length. As illustrated in Figure 6.1, the first bit is learned via:

$$\begin{aligned} \hat{b}_j^1 &= \sum_{i=1}^{N} a_j^i \varphi_1^1\left(\mathbf{x}_j^i\right), \\ b_j^1 &= \operatorname{sgn}\left(\hat{b}_1\right). \end{aligned} \tag{6.2}$$

In particular, the CNN block $\varphi_1^1()$ maps the input feature vector from $\mathcal{R}^D$ to $\mathcal{R}^1$. The attention scores a_i are used to integrate all samples within an image set into one single score. The final binary output $b_j^1 \in \{0, 1\}$ is obtained after applying the sign function $\operatorname{sgn}()$. The l^{th} bit b_j^l id learned recursively via:

$$\begin{aligned} \hat{b}_j^l &= \sum_{i=1}^{N} a_i \varphi_2^l\left(\hat{b}_j^{l-1}, \varphi_1^l\left(\mathbf{x}_j^i\right)\right), \\ b_j^l &= \operatorname{sgn}\left(\hat{b}_j^l\right). \end{aligned} \tag{6.3}$$

The input feature vectors are first embedded and integrated into a single score the same way as the first bit b_j^1. Then, the output is concatenated with the previous bit $\hat{b}_j^{l-1}$ as meta input. A second CNN block $\varphi_2^l()$ is applied on the meta input. Then, the output is rounded via the sign function. To optimize the parameters from the CNN blocks $\left\{\varphi_1^1, \varphi_1^l, \varphi_2^l\right\}_{l=2}^{L}$, an angular linear layer is added

after the binary embedding and the angular-softmax loss function is defined as,

$$\mathcal{L}_b = -\frac{1}{K}\sum_j \log\left(\frac{e^{\phi(\mathbf{b}_j,\mathbf{w}_{y_j})}}{e^{\phi(\mathbf{b}_j,\mathbf{w}_{y_j})}+e^{\sum_{c\neq y_j}\phi(\mathbf{b}_j,\mathbf{w}_c)}}\right)$$
$$\phi\left(\mathbf{b}_j,\mathbf{w}_c\right) = \|\mathbf{b}_j\|\cos\left(k\theta_{j,c}\right) \tag{6.4}$$

on a training batch $\{(\mathbf{b}_j, y_j)\,|j \in [1, K]\}$, where y_j is the class label for template $\mathbf{b}_j$. Similar with Equation (6.1.1), $\theta_{j,c}$ is the angle between the template $\mathbf{b}_j$ and projection from the c^{th} class $\mathbf{w}_c$. The $\cos\left(k\theta_{j,c}\right)$ can be represented using expressions containing $\mathbf{b}_j$ and $\mathbf{w}_c$ only via the multi-angle formula. Since the sign function is not differentiable, $\mathbf{b}_j$ is replaced with $\tanh\left(\hat{\mathbf{b}}_j\right)$. To eliminate the rounding error, a standard deviation term [18] is added to encourage the model to output binarized value:

$$\mathcal{L}_{\hat{b}} = \mathcal{L}_b + \lambda\sum_j \sigma\left(\hat{\mathbf{b}}_j\right), \tag{6.5}$$

where $\sigma\left(\right)$ denotes the standard deviation function, and $\lambda \in [0, 1]$ is a trade-off coefficient.

6.2 Implementation Details

Network Architecture: The backbone network (i.e. φ_r and φ_x) employed for ARBE is a 64-layer ResNet as described by Liu et al. [46]. The detailed architectures of the backbone network and attention block are depicted in Figure 6.2. The attention block used a fully connected layer to embed the mid-level feature representation into the attention score. Instead of generating the attention scores from the high-level feature representation [85], the output of the second ResNet block was selected, with the hypothesis tested in [47] that only identity-related information was preserved in high-level feature representation. For the recursive binary embedding layers, the first level projections $\left\{\varphi_1^l\right\}_{l=1}^L$ are 256×1 fully connected layers. The second level projections $\left\{\varphi_2^l\right\}_{l=2}^L$ are 2×1 fully connected layers.

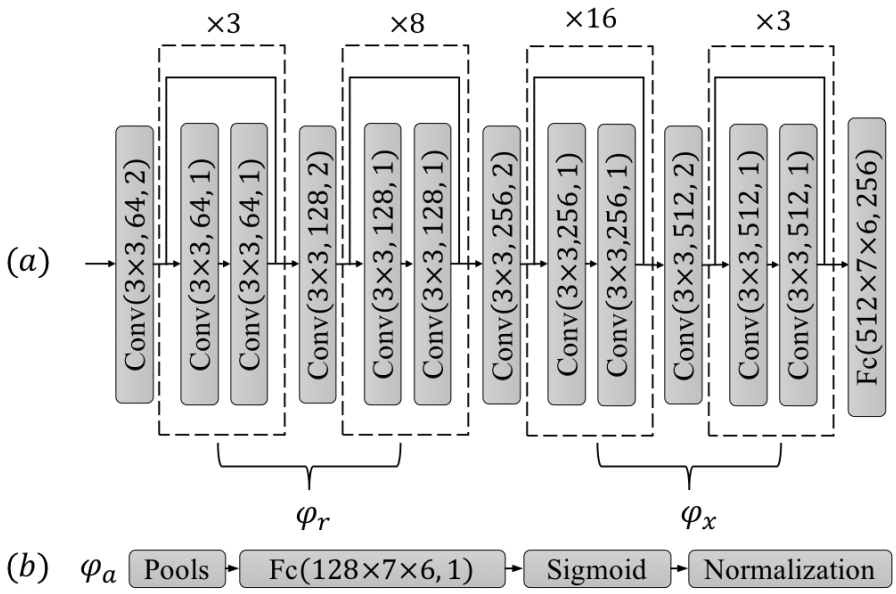

Figure 6.2: Depiction of network architectures, where Conv$(k \times k, n, s)$ denotes a $2d$ convolutional layer, using n $k \times k$ kernels with stripe s, and activated via PRelu. (a): The backbone network. (b): The attention block. The pools is two cascaded $2d$ max-pooling layers. The input of φ_a is the output from φ_r

Algorithm 6.1: Attention-based Recursive Binary Embedding

input : Training batches $\{(\mathbf{I}_j, y_j) \,|j \in [1, K]\}$

output: Network parameters in φ_r, φ_x, φ_a, and $\left\{\varphi_1^1, \varphi_1^l, \varphi_2^l\right\}_{l=2}^{L}$

```
Initialize φ_r, φ_x, φ_a;
while L_f does not converge do
    Update φ_r and φ_x
end
while L_f does not converge do
    Update φ_a
end
Initialize {φ_1^1, φ_1^l, φ_2^l}_{l=2}^L;
while L_b does not converge do
    Update {φ_1^1, φ_1^l, φ_2^l}_{l=2}^L
end
```

Training: The objective of the training stage was to optimize all the parameters from all the CNN blocks from the attention-based feature extraction network $\{\varphi_r, \varphi_a, \varphi_x\}$, and the recursive binary embedding network $\left\{\varphi_1^1, \varphi_1^l, \varphi_2^l\right\}_{l=2}^{L}$. The training steps are summarized in Algorithm 6.1. In particular, the feature extraction network $\{\varphi_r, \varphi_x\}$, the attention block$\{\varphi_a\}$, and the recursive embedding layers $\left\{\varphi_1^1, \varphi_1^l, \varphi_2^l\right\}_{l=2}^{L}$ were optimized one by one independently. In particular, $\{\varphi_r, \varphi_x\}$ was learned by minimizing Equation (6.1.1), with each sample assigned the same attention score $\varphi_a\left(x_j^i\right) = 1/N$. Then, φ_a was optimized via minimizing Equation (6.1.1), with the learned $\{\varphi_r, \varphi_x\}$. Finally, $\left\{\varphi_1^1, \varphi_1^l, \varphi_2^l\right\}_{l=2}^{L}$ is optimized via minimizing Equation (3.5).

Testing: In the testing phase, the major task was to measure the similarity between two binary templates $\mathbf{b}_j$ and $\mathbf{b}_t$. The angular similarity is defined as follows:

$$\mathcal{S}\left(\mathbf{b}_j, \mathbf{b}_t\right) = \frac{\mathbf{b}_j \mathbf{b}_t^T}{\sum_{l=1}^{L}(b_j^l + b_j^l)}. \tag{6.6}$$

6.3 Experiments

In this section, the datasets and baseline algorithms used in the experiments and the corresponding results are discussed.

6.3.1 Datasets

The CASIA WebFace Database (Webface) [87] is used for training. The Webface dataset comprises $494,414$ face images from $10,575$ subjects. Since this dataset does not provide an image set protocol, for each subject, eight images are randomly sampled as an image set to train the attention branch φ_a.

The IARPA Janus Benchmark A datasets (IJB-A)[41] is selected to assess the performance of the proposed ARBE on set-based face verification and identification. The IJB-A dataset comprises $5,397$ still images and $2,042$ videos from 500 subjects. The IJB-A is provided with a template setting, where each template contains various numbers of images from the same subjects. IJB-A also provides 10 training and testing splits. In this chapter, the training sets are not used. All the algorithms trained on Webface are tested directly to assess the generalization property across datasets. In particular, the original protocols of IJB-A are employed. There are two tasks: (i) 1:1 verification , and (ii) 1:N identification. In the first task, a list of pairs of templates is given, and the task is to distinguish whether each pair of templates describes the same subject or not. In the second task, there is a list of gallery templates and a list of probe templates. The task is to recall one or multiple templates from the gallery that are most similar to each of the probe templates.

6.3.2 Baseline Algorithms

To assess the performance of the proposed ARBE algorithm for set-based face verification, closed set identification and open set identification, the PAM [53], NAN [85], and DR-GAN [64] were selected as a baseline. They are state-of-the-art algorithms in set-based face recognition. Since the above methods generate real signatures in different dimensions, they were also compared to the state-of-the-art face retrieval algorithms, LSFS [68] and Face-int32 [18], where the binary templates were generated. In particular, the results from PAM, DR-GAN, and LSFS were copied from the original papers, because the results were reported for the same dataset using the same protocol. Although the performance on IJB-A was also reported in the original paper of NAN, the attention block was fine-tuned on IJB-A dataset. To conduct a fair comparison, NAN was implemented as described in the original paper, but used a different backbone network and a different loss function. Specifically, the backbone network and loss function are kept the same with the proposed ARBE. The Face-int32 was implemented using the same backbone network and primary loss. All algorithms were trained on Webface and tested on IJB-A with face detected with MTCNN[1] and aligned as Liu et al. [46] described.

6.3.3 Experimental Results

Experiment 1: The objective of this experiment was to assess the performance of ARBE at a code length of 256 against state-of-the-art approaches for three tasks, (i) verification, (ii) open set identification, and (iii) closed set identification. Theoretically, a 256 dimensional binary template can represent at most 1.1579209×10^{77} subjects. For verification, the true accept rates (TAR) when the false accept rate (FAR) is equal to 0.01 and 0.1 was reported. Similarly, the true positive identification rates (TPIR) when the false positive identification rate (FPIR) for the open set identification

[1] https://github.com/pangyupo/mxnet_mtcnn_face_detection

is equal to 0.01 and 0.1 were reported for open set identification. The closed set identification was conducted by removing the subjects that only appear in the probe templates. The corresponding rank-1 and rank-5 identification rates were reported. A summary of the results is provided in Table 6.1. The proposed ARBE achieves the best performance in all seven metrics compared to two other binary templates. ARBE achieved higher or comparable performance compared to the real-valued templates. To better understanding the contribution of each design component in the whole framework, additional ablation studies were performed.

Experiment 2: The objective of this experiment was to verify the contribution of different design components in ARBE framework from three aspects, (i) code structure, (iii) fusion stage, and (iv) similarity. Results are summarized in Table 6.2. In particular, in the last row, results of the proposed design is reported. On the first row, the proposed recursive embedding is compared with sequential embedding regarding different code structures. The sequential embedding was implemented using a fully connected layer (i.e. linear projections). The proposed recursive coding structure performs better than the sequential structure. On the second row, the results are fusion strategies on different stages. "Early" denotes fusion at real-valued feature level, like NAN [85] and QAN [47]. Then, the recursive embedding was learned from the fused real-valued template. The label "Late" denotes the proposed loss-level fusion. The results indicate that the latter was better. The third row represents results obtained by replacing angular similarity with Hamming distance. The performance drops significantly when using Hamming distance. On average, the impact of similarity measurements had the largest impact, and the fusion stage had the smallest impact.

Experiment 3: The objective of this experiment is to assess the performance of ARBE for different code lengths. The CMC curves for different code lengths (i.e. 256, 128, 64, and 32) are presented in Figure 6.3. The performance dropped when decreasing the code length. The 256-bit templates achieved comparable performance with the 512-bit templates, and this performance reached the

Table 6.1: Performance evaluation on the IJB-A dataset

Method	Dim.	1:1 Verification TAR(%)			1:N Identification TPIR(%)			
		FAR $= 0.001$	FAR $= 0.01$	FAR $= 0.1$	FPIR $= 0.01$	FPIR $= 0.1$	Rank 1	Rank 5
PAMs	$\mathcal{R}^{4096}$	65.2 ± 3.7	**82.6** ± 1.8	N/A	N/A	N/A	84.0 ± 1.2	92.5 ± 0.8
DR-GAN	$\mathcal{R}^{320}$	53.9 ± 4.3	77.4 ± 2.7	N/A	N/A	N/A	85.5 ± 1.3	**94.7** ± 1.1
NAN	$\mathcal{R}^{128}$	50.4 ± 8.2	74.1 ± 2.7	89.8 ± 1.0	80.2 ± 1.4	93.0 ± 0.8	82.7 ± 1.3	91.3 ± 1.3
LSFS	$\{0,1\}^{64\times8}$	51.0 ± 6.1	72.9 ± 3.5	89.3 ± 1.4	39.2 ± 2.7	61.5 ± 4.6	82.2 ± 2.3	93.1 ± 1.4
Face-int32	$\{0,1\}^{256}$	64.6 ± 3.4	80.2 ± 1.1	91.4 ± 0.6	84.2 ± 1.5	93.3 ± 0.7	85.4 ± 1.3	92.6 ± 1.1
ABRE	$\{0,1\}^{256}$	**65.6** ± 3.5	82.0 ± 1.4	**92.3** ± 0.8	**85.4** ± 1.5	**93.9** ± 1.0	**86.2** ± 1.5	92.9 ± 1.2

Table 6.2: Ablation studies on IJB-A dataset

Code	Fusion	Similarity	1:1 Verification TAR(%)		1:N Identification TPIR(%)		
Structure	Stage	Measurement	FAR = 0.01	FAR = 0.1	FPIR = 0.01	FPIR = 0.1	Rank 1
Sequential	Late	Angular	78.8 ± 1.8	91.0 ± 0.8	84.4 ± 1.5	93.4 ± 0.9	85.6 ± 1.1
Recursive	Early	Angular	77.9 ± 1.8	90.6 ± 0.7	82.7 ± 1.1	93.5 ± 0.9	85.2 ± 1.1
Recursive	Late	Hamming	64.5 ± 3.1	90.3 ± 0.9	67.4 ± 4.6	91.8 ± 1.0	76.9 ± 1.9
Recursive	Late	Angular	**82.0** ± 1.4	**92.3** ± 0.8	**85.4** ± 1.5	**93.9** ± 1.0	**86.2** ± 1.5

Delong Test on AUCs (row 4 vs. row 1): $p = 0.00003 < 0.005$

Wilcoxon signed-rank test on full-rank scores for close-set identification (row 4 vs. row 1): $p = 5.37E - 10 < 0.05$

upper bound.

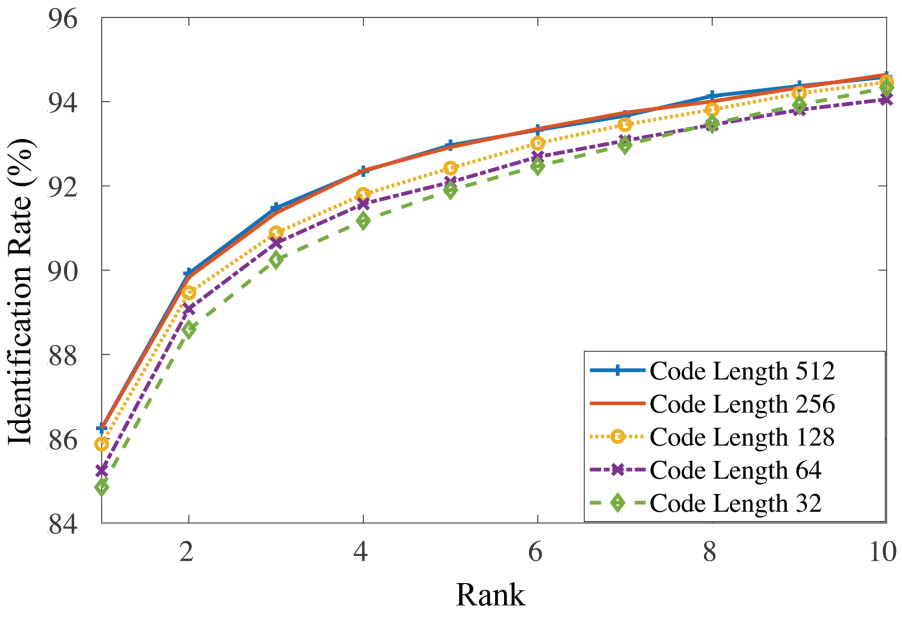

Figure 6.3: CMC curves for different code lengths.

Chapter 7

Conclusion and Future Work

This dissertation is focused on the problem of set-based face recognition. The primary contribution was achieved by advancing set-based face recognition with more compact templates and more effective matching algorithms. The problem and existing literature were analyzed and discussed to identify the challenges and limitations that restrict performance. In this dissertation, a series of algorithms were proposed to address the challenges and overcome limitations in set-based face recognition.

The SPML was first proposed to generate the gallery template with a reduced number of prototypes and learn a distance metric for similarity measurements. As demonstrated, the proposed approach can fully utilized the training data to compress the gallery image set while learning a distance metric tailored to set-to-set matching. The experimental results indicated that SPML can use a few prototypes to represent each gallery image set. Hence, it reduced the storage requirements and testing time cost while improving the identification accuracy. The corresponding sensitivity analyses indicated that SPML is robust to the number of prototypes used, the presence of outliers in the gallery and probe, as well as the prototype initialization strategy. The idea of a joint prototype

and distance metric learning for set-to-set identification can be employed in conjunction with other hull models and distance metric learning objective functions. Ablation studies on prototype learning and metric learning show that these two processes work together to improve the performance of SPML. A failure case was also investigated and an over-fitting issue was observed. Despite its many advantages, the current form of SPML can be further improved in the following aspects (i) address the over-fitting issue using regularization, (ii) extend it to unseen subjects, (iii) leverage prototype and metric learning in a different way, and (iv) embed different objective functions into this framework.

To design an algorithm with better generalization properties and take advantage of recent advances in deep learning, CDN is proposed. CDN learns a deep feature representation and a confidence score for each image in an image set. The confidence scores are the quantification of the confidence level of each image in an image set that dominates the contribution of each image to the final decision. As demonstrated, CDN improved the rank-1 identification rate for multi-probe face identification in the selected datasets. CDN exhibited superior performance when image sets contained large pose variations, whereas the improvements were not significant for image sets containing larger numbers of images. Several visual properties of the original image (e.g., pose, illumination, and skin color) were identified to affect the confidence score. However, the framework was not flexible enough with the existing sample-based FR system, joint retraining of the feature representation and confidence score are required. Moreover, the training process of CDN requires meticulous sampling for image sets and triplets. CDN can work for both verification and identification tasks. But the matching process is computationally expensive when extended to set-to-set matching.

CPN was then proposed to extend CDN into a plug-and-play module, which is more flexible with sample-based face recognition systems. CPN can be added to any sample-based face

recognition system to enhance its performance for set-based tasks without retraining the original face recognition system. Similar to CDN, it can also quantify the confidence level for each image in a set. Using the generated confidence score, it aggregates the feature representations of the original images into a single template. The training and matching processes are simplified. As demonstrated, CPN improves the state-of-the-art sample-based face recognition system in the task of set-based verification, open-set identification, and close-set identification in the selected datasets. CPN exhibits superior performance when image sets contain pose variations. Statistical tests are provided to analyze the correlations between the learned confidence scores and six visual attributes. It can be observed that the learned confidence scores show significant bias towards facial resolution, poses, gender, skin color, lighting condition, and occlusions. These biases are related to both the training data distribution and the performance bias of the backbone template. Despite its many advantages, the current form of CPN can be further improved in many aspects. First, current regression from the input image to the output performance measurements is not accurate enough. New regression models can be employed to provide better performance predictions. Second, there are many other performance measurements. Which one is more appropriate to be used to represent the confidence level? Third, current confidence scores represent the confidence level of a single sample. Is it possible to predict the confidence level of a comparison, based on both of the images to be compared?

In the end, the ARBE framework is proposed to generate a fixed-length binary template for a face image set. It combines the attention-based set aggregation used in CDN and CPN with recursive binary coding. With the same coding length, ARBE can enhance the discriminative information while restricting the number of projections used. As demonstrated, the proposed ARBE could achieve better performance on set-based face verification, open and closed set face identification compared with the state-of-the-art binary templates. The performance of ARBE is also better

or comparable with selected real-valued templates, while the template size is much smaller. Ablation studies have demonstrated that the proposed recursive coding structure, the angular similarity, and the bit-level fusion all contribute to the final improvements. Despite its many advantages, the proposed ARBE is just a first exploring of binary template for face image sets. This problem can be further analyzed in the following aspects: (i) the code structure for fast retrieval, (ii) the relationship between code length and maximum number of subjects to represent in practice, and (iii) cryptography-friendly binary template for matching.

Bibliography

[1] S. H. Abdurrahim, S. A. Samad, and A. B. Huddin. Review on the effects of age, gender, and race demographics on automatic face recognition. *The Visual Computer*, 34(11):1617–1630, 2018.

[2] G. Aggarwal, S. Biswas, P. J. Flynn, and K. W. Bowyer. Predicting good, bad and ugly match pairs. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 153–160, Breckenridge, CO, 2012.

[3] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Analysis and Applications*, 29(1):328–347, 2007.

[4] J. R. Beveridge, G. H. Givens, P. J. Phillips, and B. A. Draper. Factors that influence algorithm performance in the face recognition grand challenge. *Computer Vision and Image Understanding*, 113(6):750–762, 2009.

[5] J. R. Beveridge, G. H. Givens, P. J. Phillips, B. A. Draper, D. S. Bolme, and Y. Lui. Frvt 2006: Quo vadis face quality. *Image and Vision Computing*, 28(5):732–743, 2010.

[6] J. R. Beveridge, G. H. Givens, P. J. Phillips, B. A. Draper, and Y. Lui. Focus on quality, predicting frvt 2006 performance. In *Proc. IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–8, Amsterdam, Netherlands, 2008.

[7] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *Proc. IEEE International Conference on Computer Vision*, pages 2408–2415, Sydney, Australia, 2013.

[8] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2567–2573, San Francisco, CA, 2010.

[9] L. Chen. Dual linear regression based classification for face cluster recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2673–2680, Columbus, OH, 2014.

[10] S. Chen, C. Sanderson, M. T. Harandi, and B. C. Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 452–459, Portland, OR, 2013.

[11] Y. Chikuse. *Statistics on special manifolds, lecture notes in statistics*, volume 174. New York: Springer, 2003.

[12] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen. Image sets alignment for video-based face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2633, Providence, RI, 2012.

[13] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proc. International Conference of Machine Learning*, pages 209–216, Orlando, FL, 2007.

[14] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018.

[15] A. Deshpande, A. Shahane, D. Gadre, M. Deshpande, B. Garware, and S. Kulkarni. Towards designing an adaptive framework for facial image quality estimation at edge. In *Proc. International Conference for Convergence in Technology*, pages 1–6, Pune, India, 2018.

[16] Z. Dong, S. Jia, T. Wu, and M. Pei. Face video retrieval via deep learning of binary hash representations. In *Proc. AAAI Conference on Artificial Intelligence*, pages 3471–3477, Phoenix, AZ, 2016.

[17] A. Dutta, R. Veldhuis, and L. Spreeuwers. Predicting face recognition performance using image quality. *arXiv preprint arXiv:1510.07119*, 2015.

[18] H. Fan, M. Yang, Z. Cao, Y. Jiang, and Q. Yin. Learning compact face representation: Packing a face into an int32. In *Proc. ACM international conference on Multimedia*, pages 933–936, Orlando, FL, 2014.

[19] Q. Feng, Y. Zhou, and R. Lan. Pairwise linear regression classification for image set retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4865–4872, Las Vegas, NV, 2016.

[20] K. Grauman and R. Fergus. Learning binary hash codes for large-scale image search. In *Machine learning for computer vision*, pages 49–87, Berlin, Heidelberg, 2013.

[21] P. J. Grother, G. W. Quinn, and P. J. Phillips. Report on the evaluation of 2D still-image face recognition algorithms. *NIST interagency report*, 77(9):1–61, 2010.

[22] J. Hamm and D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proc. International Conference on Machine learning*, pages 376–383, Helsinki, Finland, 2008.

[23] M. Harandi. Dictionary learning and sparse coding on Grassmann manifolds. `https://sites.google.com/site/mehrtashharandi/publications`. Accessed: 2015-11-30.

[24] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3926–3935, Boston, MA, 2015.

[25] M. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *Proc. European Confernce on Computer Vision*, pages 17–32, Zürich, Switzerland, 2014.

[26] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *Proc. IEEE International Conference on Computer Vision*, pages 3120–3127, Sydney, Australia, 2013.

[27] M. T. Harandi, M. Salzmann, R. Jayasumana, R. Hartley, and H. Li. Expanding the family of Grassmannian kernels: An embedding perspective. In *Proc. European Confernce on Computer Vision*, pages 408–423, Zürich, Switzerland, 2014.

[28] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2705–2712, Colorado Springs, CO, 2011.

[29] M. Hayat, M. Bennamoun, and S. An. Reverse training: An efficient approach for image set classification. In *Proc. European Conference on Computer Vision*, pages 784–799, Zürich, Switzerland, 2014.

[30] M. Hayat, M. Bennamoun, and S. An. Deep reconstruction models for image set classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):713–727, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, 2016.

[32] N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear algebra and its applications*, 103:103–118, 1988.

[33] Y. Hu, A. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011.

[34] Y. Hu, A. Mian, and R. Owens. Face recognition using sparse approximated nearest points between image sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1992–2004, 2012.

[35] Z. Huang and L. J. Van Gool. A Riemannian network for SPD matrix learning. In *Proc. AAAI Conference on Artificial Intelligence*, pages 2036–2042, San Francisco, CA, 2017.

[36] Z. Huang, R. Wang, S. Shan, and X. Chen. Hybrid Euclidean-and-Riemannian metric learning for image set classification. In *Proc. Asian Conference on Computer Vision*, pages 562–577, Singapore, Singapore, 2014.

[37] Z. Huang, R. Wang, S. Shan, and X. Chen. Learning Euclidean-to-Riemannian metric for point-to-set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1677–1684, Columbus, OH, 2014.

[38] Z. Huang, R. Wang, S. Shan, and X. Chen. Projection metric learning on Grassmann manifold with application to video based face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 140–149, Boston, MA, 2015.

[39] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *Proc. International Conference on Machine Learning*, pages 1–10, Lille-Euralille, France, 2015.

[40] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008.

[41] B. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, Boston, MA, 2015.

[42] B. F. Klare and A. K. Jain. Face recognition: Impostor-based measures of uniqueness and quality. In *Proc. IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 237–244, Arlington, VA, 2012.

[43] H. Le and I. A. Kakadiaris. UHDB31: A dataset for better understanding face recognition across pose and illumination variation. In *Proc. IEEE International Conference on Computer Vision Workshops*, pages 2555–2563, Venice, Italy, 2017.

[44] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–7, Madison, WI, 2003.

[45] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen. Face video retrieval with image query via hashing across Euclidean space and Riemannian manifold. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4758–4767, Boston, MA, 2015.

[46] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6738–6746, Honolulu, HI, 2017.

[47] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Honolulu, HI, 2017.

[48] B. Lu, J.-C. Chen, C. D. Castillo, and R. Chellappa. An experimental evaluation of covariates effects on unconstrained face verification. *arXiv preprint arXiv:1808.05508*, 2018.

[49] J. Lu, G. Wang, W. Deng, and P. Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *Proc. European Conference on Computer Vision*, pages 265–280, Zürich, Switzerland, 2014.

[50] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou. Multi-manifold deep metric learning for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1137–1145, Boston, MA, 2015.

[51] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *Proc. IEEE International Conference on Computer Vision*, pages 329–336, Sydney, Australia, 2013.

[52] A. Mahmood, A. Mian, and R. Owens. Semi-supervised spectral clustering for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 121–128, Columbus, OH, 2014.

[53] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4838–4846, Las Vegas, NV, 2016.

[54] B. Maze, J. Adams, J. Duncan, N. Kalka, T. Miller, C. Otto, A. Jain, W. Niggel, J. Anderson, J. Cheney, et al. IARPA Janus Benchmark–C: Face dataset and protocol. In *Proc. International Conference on Biometrics*, pages 158–165, Queensland, Australia, 2018.

[55] B. McFee and G. R. Lanckriet. Metric learning to rank. In *Proc. International Conference on Machine Learning*, pages 775–782, Haifa, Israel, 2010.

[56] A. Mian, Y. Hu, R. Hartley, and R. Owens. Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning. *IEEE Transactions on Image Processing*, 22(12):5252–5262, 2013.

[57] P. Moutafis, M. Leng, and I. A. Kakadiaris. An Overview and Empirical Comparison of Distance Metric Learning Methods. *IEEE Transactions on Cybernetics*, 47(3):612–625, 2017.

[58] X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[59] S. Qiao, R. Wang, S. Shan, and X. Chen. Deep video code for efficient face video retrieval. In *Proc. Asian Conference on Computer Vision*, pages 296–312, Taibei, Taiwan, 2016.

[60] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, Boston, MA, 2015.

[61] G. Shakhnarovich, J. Fisher, and T. Darrell. Face recognition from long-term observations. In *Proc. European Confernce on Computer Vision*, pages 851–865, Copenhagen, Denmark, 2002.

[62] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Proc. Advances in Neural Information Processing*, pages 144–152, Stateline, NV, 2012.

[63] H. Sun, X. Zhen, Y. Zheng, G. Yang, Y. Yin, and S. Li. Learning deep match kernels for image-set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6240–6249, Honolulu, HI, 2017.

[64] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proc. IEEE International Conference on Computer Vision*, pages 1–8, Honolulu, HI, 2017.

[65] M. Uzair, A. Mahmood, and A. Mian. Sparse kernel learning for image set classification. In *Proc. Asian Conference on Computer Vision*, pages 617–631, Singapore, Singapore, 2014.

[66] R. Vemulapalli, J. K. Pillai, and R. Chellappa. Kernel learning for extrinsic classification of manifold features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1782–1789, Portland, OR, 2013.

[67] Y. Vizilter, V. Gorbatsevich, A. Vorotnikov, and N. Kostromov. Real-time face identification via CNN and boosted hashing forest. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 146–154, Las Vegas, NV, 2016.

[68] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1122–1136, 2017.

[69] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. Loy. The devil of face recognition is in the noise. In *Proc. European Conference on Computer Vision*, pages 780–795, Munich, Germany, 2018.

[70] P. Wang, Q. Ji, and J. L. Wayman. Modeling and predicting face recognition system performance based on analysis of similarity scores. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):665–670, 2007.

[71] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2496–2503, Providence, Rhode Island, 2012.

[72] R. Wang, S. Shan, X. Chen, J. Chen, and W. Gao. Maximal linear embedding for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1776–1792, 2011.

[73] R. Wang, S. Shan, X. Chen, Q. Dai, and W. Gao. Manifold-manifold distance and its application to face recognition with image sets. *IEEE Transactions on Image Processing*, 21(10):4466–4479, 2012.

[74] W. Wang. Covariance discriminant learning. `http://vipl.ict.ac.cn/homepage/rpwang/publications/CDL_Release_v1.0.rar`. Accessed: 2019-01-30.

[75] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen. Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2048–2057, Boston, MA, 2015.

[76] W. Wang, R. Wang, S. Shan, and X. Chen. Probabilistic nearest neighbor search for robust classification of face image sets. In *Proc. IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–7, Ljubljana, Slovenia, 2015.

[77] W. Wang, R. Wang, S. Shan, and X. Chen. Discriminative covariance oriented representation learning for face recognition with image sets. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5749–5758, Honolulu, HI, 2017.

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

[79] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. European Conference on Computer Vision*, pages 499–515, Amsterdam, Netherlands, Oct. 8-16 2016.

[80] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–534, Providence, RI, 2011.

[81] Y. Wu, M. Minoh, and M. Mukunoki. Collaboratively regularized nearest points for set based recognition. In *Proc. British Machine Vision Conference*, pages 1–8, Bristol, UK, 2013.

[82] Y. Wu, M. Minoh, M. Mukunoki, and S. Lao. Set based discriminative ranking for recognition. In *Proc. European Conference on Computer Vision*, pages 497–510, Firenze, Italy, 2012.

[83] W. Xie and A. Zisserman. Multicolumn networks for face recognition. *arXiv preprint arXiv:1807.09192*, 2018.

[84] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[85] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural Aggregation Network for Video Face Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Honolulu, HI, 2017.

[86] M. Yang, P. Zhu, L. Van Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. In *Proc. IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–7, Shanghai, China, 2013.

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

[88] P. Zhu. Set-to-set distance metric learning. `https://sites.google.com/site/zhupengfeifly/home/publications`. Accessed: 2015-11-30.

[89] P. Zhu, L. Zhang, W. Zuo, and D. Zhang. From point to set: Extend the learning of distance metrics. In *Proc. IEEE International Conference on Computer Vision*, pages 2664–2671, Sydney, Australia, 2013.

[90] P. Zhu, W. Zuo, L. Zhang, S. Shiu, and D. Zhang. Image set based collaborative representation for face recognition. *IEEE Transactions on Information Forensics and Security*, 9(7):1120–1132, 2014.

[91] Y. Zhu, Z. Zheng, Y. Li, G. Mu, S. Shan, and G. Guo. Still to video face recognition using a heterogeneous matching approach. In *Proc. International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, Arlington, VA, 2015.