Deep Multi-Factor Forensic Face Recognition

A Dissertation Presented

by

Zhengming Ding

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Northeastern University

Boston, Massachusetts

2018

To my family.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest and sincerest gratitude to my advisor, Prof. Yun Raymond Fu, for his continuous guidance, advice, effort, patience, and encouragement during the past five years. The strong supports from Prof. Fu lie in academic and daily aspects, even in job searching. When I was in need, he is always willing to provide the help. I am truly fortunate to have him as my advisor. This dissertation and my current achievements would not have been possible without his tremendous help.

I would also like to thank my committee members, Prof. Stratis Ioannidis and Lu Wang for their valuable time, insightful comments and suggestions ever since my PhD research proposal. I am honored to have an opportunity to work with Prof. Dy and her student to accomplish a great paper.

I would like to thank Prof. Thomas Huang, Prof. Rama Chellappa and Dr. Lei Zhang for strong supports on my faculty job searching.

In addition, I would like to thank all the members from SMILE Lab, especially my coauthors and collaborators, Prof. Ming Shao, Dr. Kong Yu, Dr. Sheng Li, Dr. Handong Zhao, Dr. Shuyang Wang, Hongfu Liu, Shuhui Jiang, Zhiqiang Tao, Yue Wu, Kai Li, Licheng Wang. For other lab members, Prof. Zhao, Dr. Jun Li, Joe, Kunpeng Li, Songyao Jiang, Bin Sun, Haiyi Mao, I also thank you very much. I have spent my wonderful four years with these excellent colleagues and left the impressive memory.

I would like to express my gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation.

Abstract of the Dissertation

Deep Multi-Factor Forensic Face Recognition

by

Zhengming Ding

Doctor of Philosophy in Electrical and Computer Engineering

Northeastern University, 2018

Dr. Yun Fu, Advisor

Forensic science is any scientific field that is applied to the field of law. Due to the popularity of the digital media carriers such as images, videos, the facial recognition becomes another important forensic technique. The major issue of forensic face recognition is the unstable system performance due to internal factor, e.g., aging, and external factors, e.g., image resolution/modality, illumination, pose. In this thesis, we investigate a theoretical framework for forensic face recognition, subject to a variety of internal/external impact factors to tackle face recognition under different views, illuminations, resolutions, modalities, periods, when probe images are captured in the surveillance environments without collaborations. Specifically, we explore two scenarios as follows.

First of all, we explore one dominant factor which hinders the forensic face recognition, which is view variance, e.g., pose and modality. Thus, we propose multi-view face recognition, which covers two settings in multi-view face recognition. On one hand, labeled data in multiple views are available in the training stage, which is the traditional multi-view learning setting. Specifically, we address the challenging but practical situation, in which the view information of the test data is unknown. On the other hand, some source views are labeled while target views are unlabeled, which satisfies transfer learning scenarios. Specifically, we explore the practical but challenging missing modality problem.

Secondly, multiple factors are modeled as the noises as a whole. On one hand, conventional auto-encoder and its variants usually involve additive noises for training data to learn robust features, which, however, did not consider the already corrupted data. We propose a novel Deep Robust Encoder (DRE) through locality preserving low-rank dictionary to extract robust and discriminative features from corrupted data. Furthermore, we fight off one-shot face recognition, where we only have one training sample for some persons, by proposing a one-shot generative model to build a more effective face recognizer.

Chapter 1

Introduction

1.1 Background

Forensic science is any scientific field that is applied to the field of law. Documents show that ever since 1970 (at least), forensic photographic comparison was introduced to US legal system to assist the analytics in different forensic fields: fingerprint comparisons, tire tread and tool mark analysis, footwear impressions. During the past decades, due to the popularity of the digital media carriers such as images, videos, the facial recognition becomes another important forensic technique. In forensic face recognition, forensic examiners carry out manually during their investigation when there is a video or image available from crime scene. Forensic examiners perform manual examination of facial images or videos for a match with huge database of mugshots. An automatic face recognition system through computer vision and pattern recognition methods may help in narrowing the suspects list. A case study about Boston bombings shows that with certain face recognition techniques, the search space of the suspects can be significantly reduced to 1/100. But until now, there is no working face recognition system that has been accepted within the judicial system. The major issue is the unstable system performance due to internal factor, e.g., aging, and external factors, e.g., image resolution/modality, illumination, pose. As a result, mug shot photos from only a few states are included in the largest government facial identification system so far, i.e., FBI next generation identification (NGI)[1], though more states agree to share the data access with FBI.

Nonetheless, the forensic facial recognition has become an indispensable analytics tool and plays irreplaceable role in the criminal events recently, to name a few:

[1] https://www.fbi.gov/about-us/cjis/fingerprints_biometrics/ngi

- In 2011 London riots, the online video website YouTube hosts many video footages of the riots, which had been recorded and uploaded by witnesses and participants. Police were able to feed the suspects' photos into through Scotland Yard's newly updated face-matching program.
- In 2013, An armed robber was claimed as the first person arrested using Chicago police's new facial recognition technology. Police obtained images of the thief from surveillance cameras. Using the NeoFace technology, the police were able to compare the photos with the department's 4.5 million mug shot and came up with suspect, who was then identified by witnesses.

Apart from a few successful cases by the law enforcement, forensic face recognition still works as the assistant tools for the manual forensic photographic comparison due to the technical limits. Thus, we aim to investigate a theoretical framework for forensic face recognition, subject to a variety of internal/external impact factors. We plan to leverage our background in multi-view/modality face recognition, robust feature learning and deep learning, to create a framework that is able to tackle face recognition under different views, illuminations, resolutions, modalities, periods, when probe images are captured in the surveillance environments without collaborations.

Among forensic photographic comparison techniques, forensic face recognition aims to identity the suspects from a huge amount gallery photos for the current probe image. The probe image could be a sketch based on the witness's descriptions, surveillance videos, or street-shot photos from the passerby. Conventionally, forensic examiners carry out manually during their investigation with the videos or images from crime scene. Recent face recognition techniques have been introduced as the auxiliary tools, which not only improve the efficiency of forensic work performed by various law enforcement agencies but will also standardize the comparison process. The major contribution of automatic face recognition is reducing the search space of suspects, and narrowing down the name list from millions of people to hundred of them. However, until now, there is no working face recognition system that has been accepted within the judicial system, even though the computer performance has already surpassed that of human being on certain face verification tasks.

It should be noted that biometric face recognition has been used for secure building access, border control, Civil ID and login verification for years; however, the application scenarios are quite different. In security applications above, the goal is to prevent incidents from occurring, while in forensic cases typically an incident has already occurred. As a result, forensic face recognition may have to compare faces captured in two distinct environment: one from CCTV with low quality, the other from mug shot with high quality. The significant difference between them, e.g., image

resolution, lighting, pose, not only fool the computer algorithms, but also challenge the investigators, even the most experienced ones.

A recent revolution in the machine learning community raises a far reaching problem, "*Will the cognition capability of computer surpass the human being in the future*." The debate attracts considerable attention, as the multi-layer neural networks/deep learning models have achieved great success in vision recognition/detection benchmark tasks. The success lies in the fact that deep structure is able to utilize considerably more training data than ever before. In forensic science, it is the *right timing to incorporate the deep learning model to improve the forensic face recognition performance*, with the availability of the large-scale training data from police department, government (mug shot, sketches), and extensive utilization of surveillance systems. *In this thesis, we will concentrate on how to design an effective deep forensic face recognition framework to tackle multiple factors that challenge the existing forensic systems.*

1.2 Related works

1.2.1 Multi-view Learning

Conventional multi-view data analysis assumes the view knowledge of the testing data is known ahead. Therefore, its setting is usually using one or more views with labels to predict another view. Those are the popular topics belonging to this scenario: cross-pose image recognition, heterogeneous image recognition, domain adaptation/transfer learning. Generally, there are three strategies: feature adaptation [1, 2, 3], classifiers adaptation [4, 5] and deep learning [6, 7, 8].

The most popular way is adapting feature space, which is usually achieved by seeking for a shared space, through either subspace learning [1, 2] or dictionary learning [3]. For example, Kan et al. proposed a discriminative multi-view analysis method by seeking multiple view-specific projections under Fisher criteria [1]. In [9], a geodesic flow kernel is implemented to learn the transitions from source and target views, and features projected to intermediate subspaces is able to represent cross-view data well. Zheng et al. presented an approach to jointly learn a set of view-specific dictionaries and a common dictionary, where view-specific features in the sparse feature spaces spanned by the view-specific dictionary set and transfer the view-shared features in the sparse feature space spanned by the common dictionary are well-aligned [3].

Many methods attempt to adjust the classifier or boundary to fit the target data. A straightforward way is to model a shared classifier among different tasks, i.e., multi-task learning. Recently,

SVM has been broadly discussed on domain adaptation problems such as remote sensing, images recognition, video analysis, where either loss function or regularizer of SVM, or both of them are reformulated according to the specific problem. Wu et al. designed a latent kernelized structural SVM for the view-invariant action recognition, which extends the kernelized structural SVM framework to include latent variables[5]. Hoffman et al. presented a novel domain transform mixture model which outperforms a single transform model when multiple domains are present, and a constrained hierarchical clustering method that successfully discovers latent domains [10].

1.2.2 Transfer Learning

Transfer learning has been witnessed as an appealing technique in many real-world applications in computer vision and pattern recognition. Specifically, transfer learning technique is designed to address the problem when the distribution of the source domain (training data) is different from that of the target domain (test data) [11]. For example, we would like to use object images from Amazon website to recognize photos captured by digital cameras of the same object categories, where the former has rich label information but the latter has fewer label knowledge. Thus, key problems turn to be adapting either source or target domain, or both of them to mitigate the distribution differences of two domains [12, 9, 13, 14].

Generally, transfer learning can be categorized according to "domains" and "tasks". More details can be referred to the excellent surveys [11]. Among them, transductive transfer learning aims to deal with the same or similar task, but has different distributions or feature spaces across two domains [13, 15, 16]. Through feature/classifier adaptation, well-learned knowledge from source domain can facilitate the unlabeled target learning. Over the past decades, a variety of transfer learning algorithms, e.g., feature adaptation [15, 17, 13, 18] and classifier adaptation [19, 20], have been proposed and achieved promising performance.

Specifically, feature adaption, one of the most popular techniques, attempts to seek a common space where source and target data could share the similar marginal distribution or conditional distribution. Along this line, subspace learning and non-linear transformation are well explored to bridge the distribution gap across two domains [13, 17]. Besides, dictionary learning strategy manages to build one common dictionary or series of dictionaries shared by source and target domains [13, 21, 18]. With such common dictionaries, the distributions divergence across two domains would be mitigated. Different constraints, e.g., Frobenius Norm, l_1-norm and low-rank constraint, would be exploited to seek the new representations for two domains so that specific properties would be

preserved during the new feature learning.

1.2.3 Deep Learning

Deep Learning has recently attracted much attention in pattern recognition and computer vision, because of its appealing performance in various tasks. Generally, deep learning tends to extract hierarchical feature representations directly from raw data, which can disentangle different explanatory factors of variation [22]. However, all these methods depend on the assumption that deep neural networks are able to learn invariant representations that are transferable across different tasks. In reality, the domain discrepancy can be alleviated, but not removed, through deep neural networks. Domain shift has posed a bottleneck to the transferability of deep networks, resulting in statistically unbounded risk for target tasks.

Among different deep structures, auto-encoder (AE) [23] has been treated as robust feature extractors or pre-training scheme in various tasks [24, 25, 26, 27, 28, 29]. Conventional AE was proposed to encourage similar or identical input-output pairs where the reconstruction loss is minimized after decoding [23]. Follow-up work with various additive noises in the input layer is able to progressively purify the data, which fulfills the purpose "denoising" against unknown corruptions in the testing data [30]. These works as well as the most recent AE variants, e.g., multi-view AE [28] and bi-shift AE [26], all assume the training data are clean, but can be intentionally corrupted. In fact, real-world data subject to corruptions such as changing illuminations, pose variations, or self-corruption do not meet the assumption above. Therefore, learning deep features from real-world corrupted data instead of intentionally corrupted data with additive noises becomes critical to build robust feature extractor that is generalized well to corrupted testing data. To the best of our knowledge, such AE based deep learning scheme has not been discussed before.

1.2.4 One-Shot Learning

One-shot face recognition is to recognize persons with only seeing them once. This problem exists in many real applications. For example, in the scenario of large-scale celebrity recognition, it naturally happens that some celebrities only have one or very limited number of images available. Another example is in the law enforcement scenario: it is usually the case that only one image of the personal ID is available for the target person.

The challenge of one-shot face recognition lies in two parts. First, a representation model is needed to transfer the face image into a discriminative feature domain. Although recent years

have witnessed great progresses in deep learning for visual recognition, computer vision systems still lack the capability of learning visual concepts from just one or a very few examples [31]. A typical solution is to leverage many images from a different group of people (we call them *base set* and name the persons with limited number of training images *low-shot set*), and train a representation model using the images from the base set to extract face features for the images in the low-shot set.

Recently, there have been many research efforts focusing on training representation models with good generalization capability. Examples include [32, 33, 34, 35] etc., where face representation model is trained and tested across different groups of persons. However, improving the generalization and capability of face representation model is still an open problem which has attracted substantial effort in the area. When the distributions of the face dataset-A and face dataset-B are very different, the representation model trained on dataset-A may not be discriminative enough on dataset-B. For example, if the data used to train a representation model do not include sufficient number of images for persons with a certain type of skin color, the trained model usually suffer from lower accuracy for those persons.

The second challenge of one-shot face recognition comes from estimating the partition for a given person in the feature space. A representation model transfers the face images of the same person into a cluster of dots in the feature space. To recognize all the faces for a given person, we need to estimate the shape, size, and location of the partition for this person in the feature space. However, with only one image (corresponding to one dot in the feature space), it is not easy to accurately and reliably estimate the distribution of the faces of the person to be recognized, which makes it challenging to estimate the boundary of the partition for this person in the feature space.

1.3 Dissertation Organization

The rest of this dissertation is organized as follows.

In chapter 2, we develop a view-invariant framework based on both subspace and deep learning to address the test multi-view data with view information unknown. Specifically, we explore multiple view-specific structures and one view-invariant structure to solve this issue.

In chapter 3, we mainly explore to utilize the knowledge of one domain to do face recognition on another domain. We consider the missing modality problem, where we have no test data available in the training stage. Specifically, we design a two-directional transfer learning framework to iteratively seek a domain-invariant feature space.

In chapter 4, we develop a deep feature learning framework aiming to seek better feature representation for face recognition. Specifically, we build a deep auto-encoder architecture by constraining the output of the decoded features to a low-rank basis, so that our designed deep architecture is able to well deal with noisy data.

In chapter 5, we target at solving one-shot face recognition, where only one training sample is available for some persons in the training stage. We first systematically review the literature of one-shot learning, then we develop a generative one-shot face recognition model by synthesizing more valid data for one-shot persons.

In chapter 6, we present a conclusion on what we propose to solve forensic face recognition in different challenges.

Chapter 2

Multi-view Face Recognition

2.1 Background

Multi-view data analysis has attracted a great deal of attention recently [1, 36, 37, 38, 39, 40, 41, 42, 43], since multi-view data are frequently seen in reality. Take face image as an example. Various viewpoints would generate cross-pose face images while different devices would generate different modalities, e.g., low-resolution face taken by a cellphone or even collected with near-infrared sensor. This results in the difficult issue that face images could be from various view-points, even heterogeneous [37, 38, 36, 44, 39]. Such data with large view divergence would result in a challenging learning problem, in which data lying in different views show a large divergence, and therefore they cannot be directly compared. In general, different views can be treated as different domains drawn from different distributions. Therefore, it is the key to adapt one view to another view to minimize the distribution divergences across them. In this paper, we mainly focus on the specific multi-view learning problem, in which data have the same feature set but different probability distributions, e.g., the multi-pose image classification and multi-modal image classification.

In general, three categories of techniques are proposed to deal with the multi-view data problems, i.e., feature adaptation [1, 2, 3], classifiers adaptation [4] and deep learning [6, 8]. Specifically, feature adaptation methods are designed to find a common view-free space, in which the multi-view data could be aligned well. While classifier adaptation approaches tend to generalize classifiers trained on some specific views to others. Deep learning algorithms focus on constructing deep structures to extract more discriminative features shared by different views to mitigate the view divergence. Our algorithm follows in feature adaptation fashion, specifically the subspace learning scenario.

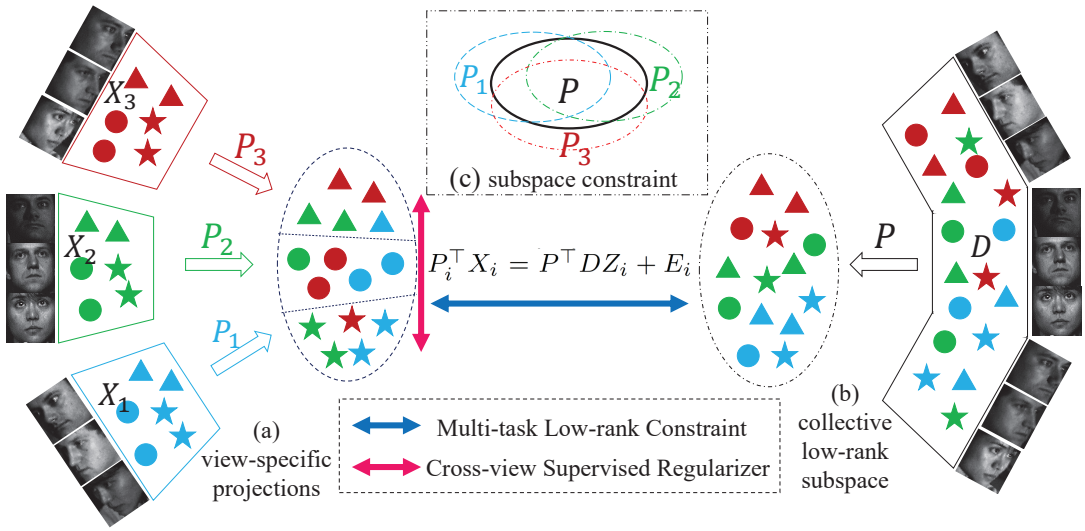


Figure 2.1: Framework of our proposed Collective Low-Rank Subspace (CLRS) algorithm.

Conventional multi-view subspace methods [1, 37] were developed to seek many view-specific projections, which transform different views into a common view-free space. Along this line, Canonical Correlation Analysis (CCA) [45] was the most representative one, which learned two projections, each for one view, to align two-view data into the shared space, respectively. Further, multi-view CCA [46] was proposed and extended to multiple view cases based on CCA. Following this, Kan et al. designed a Multi-view Discriminant Analysis (MvDA) algorithm [1], which sought an effective shared space by jointing multiple view-specific linear projections learning and Fisher constraint in a unified framework. One common drawback is that those previous researches mainly dealt with the multi-view learning tasks by applying one labeled view to predict another unlabeled view. Hence, we have to know the view knowledge of training and test data ahead of time. Only with view-information at hand can the view-specific projections be adopted to the exact views, therefore, we need a lot of prior knowledge in real-world multi-view learning scenarios.

Unfortunately, we cannot always obtain the test data's view information in advance at many real-world scenarios, since the test data are always accessible during evaluation. For example, a face image could be captured at running time with view-unknown camera so that we cannot get its exact view knowledge. In such cases, conventional multi-view learning methods cannot work, since they only built multiple view-specific projections during training stage [1, 37], which are not helpful for each view-known test data. Another phenomenon is that the test images can be in the

same distribution with the training data or totally different distributions from the training data. This leads to two scenarios: "traditional multi-view learning" and "multi-view transfer learning". When fighting off the multi-view data with no prior knowledge either view information or label knowledge or both, we can ask help from an auxiliary multi-view sources to facilitate the learning problem. In this scenario, transfer learning [11] has shown appealing performance in dealing with such a challenge. Along this line, feature adaption is a popular strategy in transfer learning, which aims to extract effective domain invariant features to reduce the domain shift so that the source knowledge could be transferred to the target [47, 16, 48].

Furthermore, low-rank modeling [49, 50] has been well exploited in transfer learning [16] and robust subspace learning [51] in the recent years. Low-rank constraint originally helps uncover the global structure of the data and detect noise or outliers. Robust subspace learning unifies low-rank modeling and dimensionality reduction to a framework by leveraging the merit of both [51], whilst low-rank transfer learning algorithms aim to uncover the intrinsic structure across source and target domains, which means each cluster in source domain is only reconstructed by one cluster in the target domain [16, 48]. To this end, marginal and conditional distribution discrepancy across source and target domains would be mitigated. Therefore, low-rank transfer learning can be an appealing data alignment tool for different distributions. In this way, low-rank reconstruction can build a bridge between view-known data and mixed view data, either in robust subspace learning or transfer learning scenario when addressing the multi-view challenge.

In this chapter, we develop a novel multi-view learning algorithm, named Collective Low-Rank Subspace (CLRS), to deal with the challenge where the view knowledge of the test data are unavailable during the learning task (Fig. 3.2). Following conventional multi-view subspace learning algorithms [1, 46, 37], we also learn the view-specific transformations for view-known training data to project the data into a latent view-free space in the training stage. Since we do not know the probe data's view information, we need to find a surrogate to preserve as much class information as possible, meanwhile reducing the impact of view divergence for mixed view-unknown test data, either in the same distribution or different distributions. On the account that the multiple view-specific projections all preserve the within-class knowledge for its specific view. In other words, those view-specific projections should have the similar discriminability for classification in different views. In other words, it is essential to find the consistent knowledge across multiple view-specific projections for view-unknown test data. This is also the core idea and uppermost contribution of this paper.

To seek a more effective projection for view-unknown test data, we employ a collective low-rank projection to uncover most of the compatible structure across multiple view-specific projections,

which are decomposed into the common part and sparse unique parts. To this end, our proposed algorithm is more flexible to solve real-world multi-view problems when we cannot have the view or even label information for the probe data at hand. Finally, we summarize our key contributions in three folds:

- A collective low-rank subspace is built through multiple view-specific projections by integrating unique parts into sparsity, so that our CLRS uncovers more shared information across different views. With low-rank constraints employed between the view-specific transformed data and commonly projected data under a multi-task scheme, our method digs out more intrinsic information by gathering cross-view data within the same class together.

- A cross-view discriminative regularizer is incorporated to align new representations of view-specific data from different tasks. This regularizer aims to align within-class data across multiple views by making full use of the label information at hand, therefore, our model can learn a more discriminative and robust collective subspace for multi-view data analysis.

- CLRS is a general method, which could be simply generalized to various learning problems (for example, conventional multi-view learning, multi-view transfer learning), by adapting different inputs. Two scenarios of experiments on several multi-view datasets are conducted to evaluate the performance of our algorithm.

2.2 Robust Multi-view Data Analysis

In this part, we briefly present our motivation, then provide our collective low-rank subspace (CLRS) for robust multi-view data analysis. Finally, we design the optimization solution and complexity analysis.

2.2.1 Motivation

In reality, multi-view data are very common and popular, since different views could facilitate the data representation, for example, near-infrared images are more insensitive to the illuminations. However, multi-view data within the same class could show definitely large diversity, resulting in a challenging issue in multi-view learning. Take multi-pose face recognition [37] as an example, we can observe that the pose variance within the subject could be extremely large so that the similarity of the same class samples in different poses would be lower than that of same pose

samples from different classes. To this end, data from the same view across different classes tend to lie in one view-specific subspace. This phenomena of multi-view data would definitely degrade the classification performance.

Research efforts on multi-view learning are exploited to seek a shared view-free representation by learning multiple view-specific projections, so that the view divergence in the observed space could be well mitigated [1, 37]. However, conventional works only build multiple view-specific projections, since they all assume the training and test data's view knowledge is already accessible. Unfortunately, we could confront such challenges in which the probe data's view knowledge is unknown ahead of time. For this reason, conventional multi-view learning algorithms cannot work in such cases, because only multiple view-specific transformations are learned, which would be invalid for the view-unknown test data.

Fortunately, data collected from various views share the same class information, therefore, each view-specific projection should have similar discriminability to separate different classes for individual view. Kan et al. mentioned that the structure of each projection for multi-view data is similar, that is view-consistency [52]. In this way, it is reasonable to consider that the multiple view-specific projection should share a lot of consistent information [22]. Furthermore, Zhu et al. provided an observation that "the identity features of the same identity are similar, even though the inputs are captured in very different views, whilst the view features of images in the same view are similar, although they are across different identities" [8].

To that end, we aim to seek a collective projection to uncover more common intrinsic knowledge across multiple views. Therefore, in this paper, we assume multiple view-specific projections share a lot of consistent information, constrained to be low-rank. Hence, the common low-rank projection can be well generalized to the view-unknown test data. Moreover, some recent works [51, 16] were designed to build a robust subspace through low-rank reconstruction to make merit from both techniques. Therefore, it is helpful to incorporate low-rank reconstruction to mitigate the distribution shift between the view-specific features and the shared features. In other words, low-rank reconstruction would guide that the view-specific features from the same class tend to be correlated to the view-free features within the same class. Such strategy would capture the global structure of the data with the help of extra prior view knowledge.

2.2.2 Conference Version Revisit

Suppose the i-th view X_i corresponds to the view-specific projection P_i and each P_i is the same size. After the projection, the data from different views would lie in a common space, so each view can span from one another. As mentioned, each view has the same class so that they should share lots of similar information within-class across different views. We assume that a low-rank common projection P can preserve this shared information, which could make the same class from different views align into the common subspace. Specifically, each P_i consists of a shared low-rank P and their unique sparse information E_i (Fig. 3.2 (c)). Therefore, we expect the common part P to be low-rank and the error E to be sparse, so more of this shared information is recoverable. We define the objective function as follows:

$$\begin{aligned} &\min_{P,E_i,P_i} \operatorname{rank}(P) + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 \\ &\text{s.t. } P_i = P + E_i, \ \ i = 1, \cdots, k, \end{aligned} \tag{2.1}$$

where $\operatorname{rank}(P)$ is the rank of matrix P, and $\|\cdot\|_1$ is l_1-norm, which is simply the maximum absolute column sum of the matrix. λ_0 is the balanced parameter between the common low-rank part and sparse ones. It is hard to directly address the rank minimization problem in Eq. (2.6). However, we are fortunately to find a good surrogate, *nuclear norm*, for the rank minimization problem [49, 50]. Therefore, Eq. (2.6) becomes:

$$\begin{aligned} &\min_{P,E_i,P_i} \|P\|_* + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 \\ &\text{s.t. } P_i = P + E_i, \ \ i = 1, \cdots, k, \end{aligned} \tag{2.2}$$

where the nuclear norm $\|\cdot\|_*$ of a matrix can be calculated by the sum of singular values of the matrix. This common low-rank P can uncover most of the shared information amongst different view-specific transformations.

Besides, low-rank representation is well-known for discovering the structure information of the data [49, 50]. Also there are several methods [53, 16, 15] that learn the low-rank structure simultaneously a robust low-dimensional subspace. Previous works have demonstrated that the low-rank structures are best uncovered jointly learning a robust low-dimensional subspace. With this idea, we aim to apply low-rank constraint to couple the view-specific transformed data and the common subspace projected data (Fig. 3.2). In detail, the data in each subspaces projected by view-specific transformations can be well reconstructed by the data lying the common subspace with

low-rank constraint. Furthermore, in real-world applications, data often include large amount of noise. Therefore, we introduce an error term to deal with noise data and get the objective function by integrating Eq. (2.5) and low-rank constraint together as:

$$\begin{aligned} &\min_{P,Z,E,E_i,P_i} \|Z\|_* + \|P\|_* + \lambda_0 \sum_{i=1}^{k} \|E_i\|_1 + \lambda_1 \|E\|_{2,1} \\ &\text{s.t. } \widetilde{X} = P^{\mathrm{T}} A Z + E, \;\; P^{\mathrm{T}} P = \mathrm{I}, \\ &P_i = P + E_i, \;\; i = 1, \cdots, k. \end{aligned} \tag{2.3}$$

where $\widetilde{X} = [P_1^{\mathrm{T}} X_1, \cdots, P_k^{\mathrm{T}} X_k]$. $\|\cdot\|_{2,1}$ is the $L_{2,1}$ norm, defined as $\|E\|_{2,1} = \sum_{k=1}^{p} \sqrt{\sum_{j=1}^{n} ([E]_{kj})^2}$, which makes it sample specific, so the outliers can be detected. And λ_1 is the balanced parameter between the error part and the low-rank part. The orthogonal constraint $P^{\mathrm{T}} P = \mathrm{I}$ is imposed to ensure the obtained P is a basis transformation matrix. Matrix A represents the data with multiple views, and is defined according to different scenarios. For representation or subspace learning, A is generally defined as the dictionary, and always replaced by the data X itself. In our experiment, we use the data itself as A for simplicity. For transfer learning, A is the target domain, while X is the source domain. It is now easier to understand, i.e. the view-information of source domain is well-learned, while the target has sparse labeled data and amount of unlabeled data. And its view information is unknown. Therefore, this formula is able to transfer the view information from the source domain to the target domain. With that common projection, we can directly extract feature from the testing data no matter what view the data are.

Up to now, we have proposed a joint learning framework by seeking the common subspace from the view-specific transformations and low-rank representation from data, simultaneously. Next, we will introduce the solution to the objective function (2.7). Following the previous multiple projections learning methods, we first transform the objective function (2.7) into the following one:

$$\begin{aligned} &\min_{P_{\mathcal{T}},P_{\mathcal{S}},Z,E,E_{\mathcal{P}}} \|Z\|_* + \|P_{\mathcal{T}}\|_* + \lambda_0 \|E_{\mathcal{P}}\|_1 + \lambda_1 \|E\|_{2,1} \\ &\text{s.t. } P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}} = P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}} Z + E, \\ &P_{\mathcal{S}} = P_{\mathcal{T}} + E_{\mathcal{P}}, \; P_{\mathcal{T}}^{\mathrm{T}} P_{\mathcal{T}} = k \cdot \mathrm{I}. \end{aligned} \tag{2.4}$$

where

$$P_{\mathcal{S}} = \begin{bmatrix} P_1 \\ \vdots \\ P_k \end{bmatrix}, \quad X_{\mathcal{S}} = \begin{bmatrix} X_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & X_k \end{bmatrix},$$

$$P_{\mathcal{T}} = \begin{bmatrix} P \\ \vdots \\ P \end{bmatrix}, \quad X_{\mathcal{T}} = \frac{1}{k} \begin{bmatrix} A \\ \vdots \\ A \end{bmatrix}, \text{ and } E_{\mathcal{P}} = \begin{bmatrix} E_1 \\ \vdots \\ E_k \end{bmatrix}.$$

Therefore, we get $\widetilde{X} = P_{\mathcal{S}}^{\mathrm{T}} X_{\mathcal{S}}$ and $P^{\mathrm{T}} A = P_{\mathcal{T}}^{\mathrm{T}} X_{\mathcal{T}}$. It is clear that $\mathrm{rank}(P) = \mathrm{rank}(P_{\mathcal{T}})$ and $\sum_{i=1}^{k} \|E_i\|_1 = \|E_{\mathcal{P}}\|_1$, so the final objective function of (2.4) can achieve the same result with (2.7). For classification tasks, we split the learned $P_{\mathcal{T}}$ into k small matrixes, then average them to achieve the common low-rank P to the final feature extraction for both gallery and probe data in the testing stage.

2.2.3 Collective Low-Rank Subspace

Assume we have k-view training data as $\mathrm{X} = [\mathrm{X}_1, \cdots, \mathrm{X}_\mathrm{k}]$, and each view $\mathrm{X}_\mathrm{i} \in \mathbb{R}^{d \times m}$ contains the same c classes with m data samples. The view-specific transformation $\bar{\mathbf{P}}_\mathrm{i} \in \mathbb{R}^{d \times d}$ would be learned for the i-th view X_i following the conventional multi-view learning [1]. Hence, each $\bar{\mathbf{P}}_\mathrm{i}$ represents the basis to expand the space of each view X_i, i.e., $\bar{\mathbf{P}}_\mathrm{i} = \mathrm{X}_\mathrm{i}\mathrm{A}_\mathrm{i}$, where A_i is weight matrix [52]. As discussed before, multiple view-specific projections have the similar discriminability in their own view so that they should have a lot of shared knowledge. Then, we aim to seek as many common bases as possible across multi-view data so that such common basis can be generalized to view-unseen test data. To this end, we adopt a collective low-rank transformation $\bar{\mathbf{P}} \in \mathbb{R}^{d \times d}$ to uncover such consistent knowledge that it can be extended to work for view-unknown test data. Specifically, we exploit low-rank sparse decomposition by assuming each $\bar{\mathbf{P}}_\mathrm{i}$ is combined of $\bar{\mathbf{P}}$ and their unique sparse residue $\bar{\mathbf{S}}_\mathrm{i} \in \mathbb{R}^{d \times d}$, so more common knowledge could be uncovered. Finally, the objective function for low-rank sparse decomposition is defined in the following:

$$\begin{aligned} &\min_{\bar{\mathbf{P}}, \bar{\mathbf{S}}_\mathrm{i}, \bar{\mathbf{P}}_\mathrm{i}} \mathrm{rank}(\bar{\mathbf{P}}) + \lambda_0 \sum_{i=1}^{k} \|\bar{\mathbf{S}}_\mathrm{i}\|_1, \\ &\text{s.t. } \bar{\mathbf{P}}_\mathrm{i} = \bar{\mathbf{P}} + \bar{\mathbf{S}}_\mathrm{i}, \ \ i = 1, \cdots, k, \end{aligned} \tag{2.5}$$

in which $\mathrm{rank}(\cdot)$ denotes the rank operator of a matrix, while $\|\cdot\|_1$ is l_1-norm that calculates the maximum absolute column sum of a matrix. $\lambda_0 > 0$ is the trade-off to balance two parts. So far, we seek a low-rank common basis without dimensionality reduction. That is, we find all the d bases for feature learning.

Remark: There are already research activities on low-rank sparse decomposition, for example, Xia et al. aimed to seek an optimal low-rank clustering matrix from multiple clustering results [54]. Although the objective function is similar, the methodology, technical idea and applications behind are very different. Specifically, Xia et al. first achieved multiple clustering results for each view, then they assumed such multiple clustering results could generate a low-rank common clustering result, which is treated as the final optimal result. Therefore, they adopted low-rank sparse decomposition to multiple clustering results to obtain the low-rank optimal one. However, we assume multiple view-specific transformations share a lot of information so that we adopt low-rank sparse decomposition to seek a collective low-rank subspace to address the challenge that the view knowledge of multi-view test data is unavailable. Considering view-consistency in multi-view data, the collective low-rank projection can capture most common structure shared by multiple view-specific transformations. That is, the deviation error matrix tends to be sparse. Furthermore, we also evaluate different types of error, e.g., sparse norm and Frobenius norm, and we found that the results are almost the same. That is, even we don't know what types of the error across multiple views, we can still apply sparse norm to model it.

2.2.4 Multi-view Low-rank Subspace Learning

Since $\bar{\mathbf{P}}$ is low-rank, there are many bases very similar, resulting in much redundant information within $\bar{\mathbf{P}}$. Assume the rank of $\bar{\mathbf{P}}$ is p ($p \ll d$), hence, we can adopt the p bases to extract effective features from multi-view data, which could help well deal with the *curse of dimensionality* [55]. Hence, we could transform the original problem into a fixed rank problem as:

$$\begin{aligned} &\min_{\mathrm{P},\mathrm{S_i},\mathrm{P_i}} \lambda_0 \sum_{i=1}^{k} \|\mathrm{S_i}\|_1, \\ &\text{s.t. } \mathrm{P_i} = \mathrm{P} + \mathrm{S_i},\ i = 1, \cdots, k,\ \mathrm{P}^\top \mathrm{P} = \mathrm{I}_p, \end{aligned} \tag{2.6}$$

where $\mathrm{P} \in \mathbb{R}^{d\times p}, \mathrm{P_i} \in \mathbb{R}^{d\times p}, \mathrm{S_i} \in \mathbb{R}^{d\times p}$ are p columns of $\bar{\mathbf{P}}, \bar{\mathbf{P}}_\mathrm{i}, \bar{\mathbf{S}}_\mathrm{i}$, respectively and we add an orthogonal constraint $\mathrm{P}^\top\mathrm{P} = \mathrm{I}_p$ ($\mathrm{I}_p \in \mathbb{R}^{p\times p}$ is an identity matrix) to make the P with the full rank of p.

Recently, low-rank modeling has been broadly studied to the global structure with the multi-class data [49, 50]. Furthermore, low-rank representation has been integrated into subspace learning framework to build a robust subspace for effective feature learning [51, 16]. Following the idea of low-rank subspace learning, we desire to exploit low-rank representation to build a bridge across the view-specific features and the shared features (Fig. 3.2). Hence, knowledge across multiple view-specific transformations could be transferred to the common subspace. Due to the real-world data are always noisy, we design a sparse error term to figure out the noise or outliers. Finally, the objective function can be achieved by integrating Eq. (2.6) and low-rank reconstruction into a unified framework as:

$$\begin{aligned} &\min_{\mathrm{P},\mathrm{Z_i},\mathrm{E_i},\mathrm{S_i},\mathrm{P_i}} \sum_{i=1}^{k} \left(\mathrm{rank}(\mathrm{Z_i}) + \lambda_0 \|\mathrm{S_i}\|_1 + \lambda_1 \|\mathrm{E_i}\|_{2,1}\right) \\ &\text{s.t. } \mathrm{P_i}^\top \mathrm{X_i} = \mathrm{P}^\top \mathrm{D} \mathrm{Z_i} + \mathrm{E_i},\ \mathrm{P}_i = \mathrm{P} + \mathrm{S}_i, \\ &i = 1, \cdots, k,\ \ \mathrm{P}^\top \mathrm{P} = \mathrm{I}_p, \end{aligned} \tag{2.7}$$

where $\mathrm{Z_i} \in \mathbb{R}^{\bar{m} \times m}$ is the i-th low-rank reconstruction coefficient. $\mathrm{E_i} \in \mathbb{R}^{p \times m}$ is the error term and $\|\cdot\|_{2,1}$ is the $L_{2,1}$-norm, i.e., $\|\mathrm{E_i}\|_{2,1} = \sum_{k=1}^{p} \sqrt{\sum_{j=1}^{m} ([\mathrm{E_i}]_{kj})^2}$, which aims to detect and remove outliers. And $\lambda_1 > 0$ is the trade-off to balance two parts. With the collective low-rank projection, we can alleviate the multi-view learning by extracting effective features for the test data whatever view the data are. Since rank minimization problem is an NP-hard problem in Eq. (2.7), recent researches adopt *nuclear norm* as a good surrogate[49, 50].

In the above objective function, $\mathrm{D} \in \mathbb{R}^{d \times \bar{m}}$ denotes the data with mixed k views, which has different definitions in different scenarios. In feature learning setting, D means the dictionary ($\bar{m}$ is the atom size of dictionary) [49, 50], which usually adopts the data itself X for simplicity. In this paper, we also directly use X as the basis. Whilst in transfer learning setting, D denotes the unlabeled target domain and X represents the well-labeled source domain. We can easily understand that we are dealing with an unlabeled multi-view dataset by borrowing the knowledge from a well-learned source domain. Objective function (2.7) would help facilitate the target learning with the the view/label knowledge of source domain.

2.2.5 Supervised Cross-view Alignment

To better utilize the label information in the training stage, we employ a supervised graph regularizer to align cross-view data within the same class. Model (2.7) only utilizes the view-information of the training data so that it works in a same weakly supervised fashion to our previous work [2]. Moreover, model (2.7) exploits a multi-task scheme, that is, data from each view are

reconstructed by the commonly projected data in an individual manner. It is very important to align different views to make the learned collective subspace more discriminative. We first denote the projected low-dimensional data of each view $\mathrm{Y}_i = \mathrm{P}_i^\top \mathrm{X}_i$ ($\mathrm{P}^\top \mathrm{D}\mathrm{Z}_i \in \mathbb{R}^{p\times m}$ can be treated as its clean version), so the multi-view projected data $\mathrm{Y} = [\mathrm{Y}_1, \cdots, \mathrm{Y}_k] \approx \mathrm{P}^\top \mathrm{D}\mathrm{Z} = \mathrm{P}^\top \mathrm{D}[\mathrm{Z}_1, \cdots, \mathrm{Z}_k] \in \mathbb{R}^{p\times km}$.

Since the fact that data from multiple views are from c different classes, these samples should be lying in c different subspaces. Hence, each view coefficient matrix Z_i tends to be low-rank. Namely, the coefficient vectors within each view corresponding to samples within the same class should be highly correlated. For multi-view learning, it is of great importance to couple within-class data across different views. We design a supervised regularization $\Omega(\mathrm{P}, \mathrm{Z})$ based on Fisher criterion as:

$$\Omega(\mathrm{P}, \mathrm{Z}) = \frac{\mathrm{tr}(\mathcal{S}_w)}{\mathrm{tr}(\mathcal{S}_b)}, \tag{2.8}$$

where $\mathrm{tr}(\mathcal{M})$ is the trace of matrix $\mathcal{M}$. $\mathcal{S}_w$ and $\mathcal{S}_b$ are the within-class and between-class scatter matrices on $\mathrm{P}^\top \mathrm{D}\mathrm{Z}$ respectively defined as:

$$\begin{aligned}
\mathcal{S}_w &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (y_j^i - \mu_i)(y_j^i - \mu_i)^\top, \\
\mathcal{S}_b &= \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^\top,
\end{aligned}$$

in which μ_i is the mean of the i-th class in Y, μ is the overall mean of Y, n_i is the size of the i-th class, and y_j^i is the j-th data in the i-th class of Y. With Fisher criterion, the low-dimensional cross-view data from different classes should be far apart, while those from the same class should be close to each other. To better solve this problem, we convert trace-ratio problem into a trace difference problem [56]. Furthermore, we involve a regularization term to guarantee the convexity of $\Omega(\mathrm{P}, \mathrm{Z})$ to Z [51] and reformulate Eq. (2.8) as:

$$\begin{aligned}
\Omega(\mathrm{P}, \mathrm{Z}) =& \mathrm{tr}(\mathcal{S}_w) - \mathrm{tr}(\mathcal{S}_b) + \eta \|\mathrm{P}^\top \mathrm{D}\mathrm{Z}\|_{\mathrm{F}}^2 \\
=& \mathrm{tr}\Big((\mathrm{P}^\top \mathrm{D}\mathrm{Z})(\mathrm{I}_{km} - \mathcal{L}_w)(\mathrm{P}^\top \mathrm{D}\mathrm{Z})^\top\Big) \\
& - \mathrm{tr}\Big((\mathrm{P}^\top \mathrm{D}\mathrm{Z})\mathcal{L}_b(\mathrm{P}^\top \mathrm{D}\mathrm{Z})^\top\Big) + \eta \|\mathrm{P}^\top \mathrm{D}\mathrm{Z}\|_{\mathrm{F}}^2 \\
=& \mathrm{tr}\big((\mathrm{P}^\top \mathrm{D}\mathrm{Z})((1+\eta)\mathrm{I}_{km} - \mathcal{L}_w - \mathcal{L}_b)(\mathrm{P}^\top \mathrm{D}\mathrm{Z})^\top\big),
\end{aligned} \tag{2.9}$$

where η is usually a small positive value (Generally, we set $\eta = 10^{-3}$) and $\|\cdot\|_{\mathrm{F}}^2$ is the matrix Frobenius norm. $\mathrm{I}_{km} \in \mathbb{R}^{km\times km}$ is an identity matrix. The elements of $\mathcal{L}_w, \mathcal{L}_b$ are defined as:

$$\mathcal{L}_w[i,j] = \begin{cases} \frac{1}{n_c}, & \text{if } y_i \text{ and } y_j \text{ belong to class } c \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{L}_b[i,j] = \begin{cases} \frac{1}{n_c} - \frac{1}{km}, & \text{if } y_i \text{ and } y_j \text{ belong to class } c \\ -\frac{1}{km}, & \text{otherwise} \end{cases}$$

To sum up, we come up with the final objective function for multi-view data analysis as:

$$\begin{aligned} \min_{\substack{\mathrm{P},\mathrm{Z_i},\mathrm{E_i},\\ \mathrm{S_i},\mathrm{P_i}}} & \sum_{i=1}^{k} \left(\|\mathrm{Z_i}\|_* + \lambda_0 \|\mathrm{S_i}\|_1 + \lambda_1 \|\mathrm{E_i}\|_{2,1} \right) + \lambda_2 \Omega(\mathrm{P}, \mathrm{Z}) \\ & \text{s.t. } \mathrm{P_i^\top X_i} = \mathrm{P^\top D Z_i} + \mathrm{E_i},\ \mathrm{P_i} = \mathrm{P} + \mathrm{S_i}, \\ & i = 1, \cdots, k,\ \ \mathrm{P^\top P} = \mathrm{I}_p, \end{aligned} \tag{2.10}$$

where $\lambda_2 > 0$ is a trade-off parameter to balance the weakly supervised multi-view parts (Eq. (2.7)) and discriminative terms (Eq. (2.9)). $\|\cdot\|_*$ denotes the nuclear norm of a matrix, which calculates the sum of singular values of a matrix. To sum up, we design a unified multi-view learning framework by jointly building a discriminative collective subspace from multiple view-specific transformations and uncovering the data's global structure through low-rank reconstruction.

Discussion: Current multi-view learning methods [1, 37, 52] adopted multiple view-specific transformations to project the original data into a view-free space so that the view divergence across different views would be mitigated. It can be observed that such multiple view-specific projections are designed to preserve the similar discriminability for the same class across different views, therefore, there exists a lot of common knowledge across multiple view-specific projections, which are independent of view variance [22]. In this paper, we adopt low-rank decomposition to seek a low-rank common projection, which aims to uncover most shared knowledge across different view-specific projections. Our collective low-rank projection could make our model more flexible in handling the challenges that we cannot achieve the view information of the probe data. Furthermore, a cross-view alignment term is incorporated into our framework to make our collective subspace more discriminative and robust.

We further discuss two most low-rank subspace learning algorithms, which are SRRS [51], LTSL [16]. Specifically, SRRS focuses on seeking robust and effective features by integrating low-rank representation and LDA-like discriminative regularizer in to a unified framework. Differently, LTSL adopts low-rank subspace for transfer learning, aiming to transfer well-labeled knowledge from source data to the target one through locality aware reconstruction. LTSL aims to seek a domain-invariant subspace by adapting the knowledge of source to the target. Furthermore, LTSL incorporates general subspace learning algorithms, e.g., PCA, LDA and LPP into the transfer learning framework. Compared with SRRS, LTSL and our CLRS both exploit the low-rank constraint using the low-dimensional features, thus, we could save computational cost during the model training.

Moreover, our algorithm is developed to deal with the challenging problem where the prior view information for the testing data is unavailable.

2.2.6 Solving Objective Function

Since objective function (2.10) has many variables to be optimized, we adopt the popular Alternating Direction Method of Multipliers (ADMM) algorithm [57] to solve the objective function (2.10). Recent researches show that ADMM converges well even with some variables non-smooth. First of all, we introduce several relaxation variables $\mathrm{J_i}, \mathrm{Q_i}$ and then reformulate Eq. (2.10) into its equivalent minimization problem as:

$$\begin{aligned} \min_{\substack{\mathrm{P},\mathrm{Z},\mathrm{E_i},\mathrm{S_i},\\ \mathrm{P_i},\mathrm{J_i},\mathrm{Z_i},\mathrm{Q_i}}} & \sum_{i=1}^{k} \big(\|\mathrm{J}_i\|_* + \lambda_0\|\mathrm{S_i}\|_1 + \lambda_1\|\mathrm{E_i}\|_{2,1}\big) + \lambda_2\Omega(\mathrm{P},\mathrm{Z}) \\ & \text{s.t. } \mathrm{P_i}^\top\mathrm{X_i} = \mathrm{P}^\top\mathrm{DQ_i} + \mathrm{E_i},\ \mathrm{P_i} = \mathrm{P} + \mathrm{S_i},\ \mathrm{Z_i} = \mathrm{J_i}, \\ & \mathrm{Z_i} = \mathrm{Q_i},\ \mathrm{P}^\top\mathrm{P} = \mathrm{I}_p,\ i = 1, \cdots, k, \end{aligned} \tag{2.11}$$

whose augmented Lagrangian function is

$$\begin{aligned} \sum_{i=1}^{k} \Big(& \|\mathrm{J_i}\|_* + \lambda_0\|\mathrm{S_i}\|_1 + \lambda_1\|\mathrm{E_i}\|_{2,1} + \langle \mathrm{U_i}, \mathrm{Z_i} - \mathrm{J_i}\rangle \\ & + \langle \Upsilon_i, \mathrm{P_i}^\top\mathrm{X_i} - \mathrm{P}^\top\mathrm{DQ_i} - \mathrm{E_i}\rangle + \langle \mathrm{V_i}, \mathrm{P_i} - \mathrm{P} - \mathrm{S_i}\rangle \\ & + \langle \mathrm{R_i}, \mathrm{Z_i} - \mathrm{Q_i}\rangle + \frac{\mu}{2}\big(\|\mathrm{P_i}^\top\mathrm{X_i} - \mathrm{P}^\top\mathrm{DQ_i} - \mathrm{E_i}\|_\mathrm{F}^2 \\ & + \|\mathrm{P_i} - \mathrm{P} - \mathrm{S_i}\|_\mathrm{F}^2 + \|\mathrm{Z_i} - \mathrm{Q_i}\|_\mathrm{F}^2 + \|\mathrm{Z_i} - \mathrm{J_i}\|_\mathrm{F}^2\big)\Big) \\ & + \lambda_2 \mathrm{tr}\Big((\mathrm{P}^\top\mathrm{DZ})\mathcal{L}(\mathrm{P}^\top\mathrm{DZ})^\top\Big), \end{aligned}$$

where $\Upsilon_i, \mathrm{U_i}, \mathrm{R_i}$, and V_i are Lagrange multipliers and μ is the positive penalty parameter. $\mathcal{L} = (1+\eta)\mathrm{I}_{km} - \mathcal{L}_w - \mathcal{L}_b$. $\langle , \rangle$ denotes the inner product operator of two matrices.

As it can be seen, it is hard to jointly update the variables in objective function (2.11). Fortunately, we can achieve the optimization solution by iteratively updating each variable. Specifically, we alternately optimize the following variables $\mathrm{J_i}, \mathrm{Z_i}, \mathrm{Q_i}, \mathrm{S_i}, \mathrm{E_i}, \mathrm{P_i}, \mathrm{S_i}, \mathrm{Z}$, and P in a leave-one-out strategy. Moreover, assume $\mathrm{J_{i,t}}, \mathrm{Z_{i,t}}, \mathrm{Q_{i,t}}, \mathrm{E_{i,t}}, \mathrm{P_{i,t}}, \mathrm{P_t}, \mathrm{S_{i,t}}, \mathrm{Z_t}, \Upsilon_{\mathrm{i,t}}, \mathrm{R_{i,t}}, \mathrm{V_{i,t}}, \mathrm{U_{i,t}}$ and μ_t are the solutions of the t-th iteration, hence the solutions in the t+1 iteration are shown as follows:

Updating $\mathrm{J_i}$:

$$\mathrm{J_{i,t+1}} = \arg\min_{\mathrm{J_i}} \frac{1}{\mu_t}\|\mathrm{J_i}\|_* + \frac{1}{2}\|\mathrm{J_i} - (\mathrm{Z_{i,t}} + \frac{\mathrm{U}_{i,t}}{\mu_t})\|_\mathrm{F}^2. \tag{2.12}$$

Updating $\mathrm{Q_i}$:

$$\mathrm{Q_{i,t+1}} = (\mathrm{D}^\top\mathrm{P_t}\mathrm{P_t}^\top\mathrm{D} + \mathrm{I}_{\bar{\mathrm{m}}})^{-1}\mathcal{Q}_{\mathrm{i,t}}, \tag{2.13}$$

where $\mathcal{Q}_{i,t} = D^\top P_t(P_{i,t}^\top X_i - E_{i,t}) + Z_{i,t} + \frac{D^\top P_t \Upsilon_{i,t} + R_{i,t}}{\mu_t}$ and $I_{\bar{m}} \in \mathbb{R}^{\bar{m}\times\bar{m}}$ is an identity matrix.

Updating E_i:

$$\begin{aligned} E_{i,t+1} = \ & \arg\min_{E_i} \tfrac{\lambda_1}{\mu_t}\|E_i\|_{2,1} \\ & + \tfrac{1}{2}\|E_i - (P_{i,t}^\top X_i - P_t^\top D Q_{i,t+1} + \tfrac{\Upsilon_{i,t}}{\mu_t})\|_F^2. \end{aligned} \tag{2.14}$$

Updating P_i:

$$\begin{aligned} P_{i,t+1} = \ & (X_i X_i^\top + I_d)^{-1}\Big(X_i(Q_{i,t+1}^\top D^\top P_t + E_{i,t+1}^\top) \\ & + P_t + S_{i,t+1} - \tfrac{X_i \Upsilon_{i,t}^\top + V_{i,t}}{\mu_t}\Big). \end{aligned} \tag{2.15}$$

Updating S_i:

$$S_{i,t+1} = \arg\min_{S_i} \tfrac{\lambda_0}{\mu_t}\|S_i\|_1 + \tfrac{1}{2}\|S_i - (P_{i,t+1} - P_t + \tfrac{V_{i,t}}{\mu_t})\|_F^2. \tag{2.16}$$

Updating P:

$$P_{t+1} = \arg\min_{P} \mathcal{F}(P),\ \text{s.t. } P^\top P = I_p, \tag{2.17}$$

where $\mathcal{F}(P) = \sum_{i=1}^{K} \frac{\mu_t}{2}\big(\|P_{i,t+1} - P - S_{i,t+1} + \frac{V_{i,t+1}}{\mu_t}\|_F^2 + \|P_{i,t+1}{}^\top X_i - P^\top D Q_{i,t+1} - E_{i,t+1} + \frac{\Upsilon_{i,t}}{\mu_t}\|_F^2\big) + \lambda_2 \mathrm{tr}\big((P^\top D Z_{t+1})\mathcal{L}(P^\top D Z_{t+1})^\top\big)$ and this optimization is a non-convex problem. To address the difficult non-convex problem (2.17) due to the orthogonal constrains, we use a gradient descent optimization procedure with curvilinear search for a local optimal solution and readers can refer to [58] for details. Generally, we first calculate the gradient of $\mathcal{F}(P)$ w.r.t P as $\frac{\partial \mathcal{F}(P)}{\partial P} = \mu_t\Big(P - (P_{i,t+1} - S_{i,t+1} + \frac{V_{i,t+1}}{\mu_t}) + (DQ_{i,t+1})(Q_{i,t+1}^\top D^\top P - (P_{i,t+1}{}^\top X_i - E_{i,t+1} + \frac{\Upsilon_{i,t}}{\mu_t})^\top\Big) + 2\lambda_2 D Z_{t+1}\mathcal{L}(D Z_{t+1})^\top P$. Then we calculate skew-symmetric matrix and optimize P until Armijo-Wolfe conditions [59] meet.

Updating Z:

$$\begin{aligned} & Z_{t+1} + \lambda_2(D^\top P_{t+1} P_{t+1}^\top D) Z_{t+1}\mathcal{L} - G_t = 0 \\ \Rightarrow\ & Z_{t+1}\mathcal{L}^{-1} + \lambda_2(D^\top P_{t+1} P_{t+1}^\top D) Z_{t+1} = G_t \mathcal{L}^{-1}, \end{aligned} \tag{2.18}$$

where $G_t = [G_{1,t}, \cdots, G_{k,t}]$ and $G_{i,t} = \frac{1}{2}(Q_{i,t+1} + J_{i,t+1} - \frac{U_{i,t} + R_{i,t}}{\mu_t})$. When Z_{t+1} is learned, we could split it into $Z_{i,t+1}$.

Specifically, Eq. (2.18) is a standard Sylvester equation, and can be effectively solved via Matlab function. Eq. (3.16) is for nuclear-norm, which can be solved by Singular Value Thresholding (SVT) [60]. Eqs. (2.14)(2.16) are two sparse problems, which can be solved by the popular soft-thresholding operator [61]. For clarity, we list the detailed steps of the optimization in **Algorithm 1**.

So far, it is still difficult to guarantee the convergence of ADMM with three or more blocks [50]. Recent research efforts are exploited to prove the convergence for non-smooth and non-convex

Algorithm 1 Solving Problem (2.11) by ADMM

Input: $X_i (i = 1, \cdots, k), D, \lambda_0, \lambda_1, \lambda_2$

Initialize: $\mathrm{J_{i,0}} = \mathrm{Q_{i,0}} = \mathrm{E_{i,0}} = \mathrm{S_{i,0}} = \Upsilon_{\mathrm{i,0}} = \mathrm{U_{i,0}} = \mathrm{V_{i,0}} = 0$, $\mathrm{Z}_0 = 0, \mu_0 = 10^{-5}, \rho = 1.3, \max_\mu = 10^8, \epsilon = 10^{-5}, t = 0$.

while not converged **do**

1. Optimize $\{\mathrm{J/Q/E/P/S}\}_{\mathrm{i,t+1}}$ in parallel.
   - 1-1 Optimize $\mathrm{J_{i,t+1}}$ via Eq. (3.16) by fixing others.
   - 1-2 Optimize $\mathrm{Q_{i,t+1}}$ via Eq. (3.17) by fixing others.
   - 1-3 Optimize $\mathrm{E_{i,t+1}}$ via Eq. (2.14) by fixing others.
   - 1-4 Optimize $\mathrm{P_{i,t+1}}$ via Eq. (2.15) by fixing others.
   - 1-5 Optimize $\mathrm{S_{i,t+1}}$ via Eq. (2.16) by fixing others.
2. Optimize $\mathrm{P_{t+1}}$ via Eq. (2.17) by fixing others.
3. Optimize $\mathrm{Z_{t+1}}$ via Eq. (2.18) by fixing others.
4. Optimize the multipliers $\Upsilon_{\mathrm{i,t+1}}, \mathrm{U_{i,t+1}}, \mathrm{V_{i,t+1}}, \mathrm{R_{i,t+1}}$ via
   - $\Upsilon_{\mathrm{i,t+1}} = \Upsilon_{\mathrm{i,t}} + \mu_\mathrm{t}(\mathrm{P_{i,t+1}^\top X_i} - \mathrm{P_{t+1}^\top DQ_{i,t+1}} - \mathrm{E_{i,t+1}})$;
   - $\mathrm{U_{i,t+1}} = \mathrm{U_{i,t}} + \mu_\mathrm{t}(\mathrm{Z_{i,t+1}} - \mathrm{J_{i,t+1}})$;
   - $\mathrm{R_{i,t+1}} = \mathrm{R_{i,t}} + \mu_\mathrm{t}(\mathrm{Z_{i,t+1}} - \mathrm{Q_{i,t+1}})$;
   - $\mathrm{V_{i,t+1}} = \mathrm{V_{i,t}} + \mu_\mathrm{t}(\mathrm{P_{i,t+1}} - \mathrm{P_{t+1}} - \mathrm{S_{i,t+1}})$;
5. Optimize μ_{t+1} via $\mu_{t+1} = \min(\rho\mu_t, \max_\mu)$.
6. Check the convergence conditions
   - $\|\mathrm{P_{i,t+1}^\top X_i} - \mathrm{P_{t+1}^\top DQ_{i,t+1}} - \mathrm{E_{i,t+1}}\|_\infty < \epsilon$;
   - $\|\mathrm{Z_{i,t+1}} - \mathrm{J_{i,t+1}}\|_\infty < \epsilon, \|\mathrm{Z_{i,t+1}} - \mathrm{Q_{i,t+1}}\|_\infty < \epsilon$;
   - $\|\mathrm{P_{i,t+1}} - \mathrm{P_{t+1}} - \mathrm{S_{i,t+1}}\|_\infty < \epsilon$.
7. $t = t + 1$.

end while

output: $\mathrm{Z_i}, \mathrm{Q_i}, \mathrm{J_i}, \mathrm{E_i}, \mathrm{S_i}, \mathrm{P_i}, \mathrm{P}, \mathrm{Z}$.

optimization problem [62, 63]. However, our problem is much more complex, especially, we have an orthogonal constraint on P, so we did not prove the convergence theoretically. We will show the convergence analysis in the experimental part. Furthermore, the parameters μ_0, ρ, ϵ, η, and $\max_\mu$ are set empirically, while the three trade-off parameters $\lambda_0, \lambda_1, \lambda_2$ and p are tuning throughout the experiment. Besides, $\mathrm{P_i}$ is initialized with PCA [64] for each view data $\mathrm{X_i}$ and P is initialized with PCA on all the data.

Discussion: We propose a different objective function and optimization scheme, comparing with our conference version [2], which applies the low-rank constraint as $\widetilde{\mathrm{X}} = \mathrm{P}^\top \mathrm{DZ} + \mathrm{E}$ and $\widetilde{\mathrm{X}} = [\mathrm{P_1^\top X_1}, \cdots, \mathrm{P_k^\top X_k}]$. In [2], we assume the new representations Z of multi-view data together to be low-rank, while here we constrain the new representation Z_i of each view data to be low-rank. In our previous conference

version [2], we stack multiple view-specific projections to a large new projection, so is the collective projection. When we employ low-rank constraint on the common subspace, we previously would need much more computational time and space than our current one.

Furthermore, we adopt a multi-task low-rank framework to reconstruct view-specific projected data with mixed-view data in the collective subspace. Hence, variables $\mathrm{J_i}, \mathrm{Q_i}, \mathrm{E_i}, \mathrm{S_i}, \mathrm{P_i}$ in each task can be solved in a parallel scheme, i.e., Step 1-1 to 1-5. Therefore, we can save much time to solve those variables with new parallel techniques. Besides, we relax the low-rank constraint on big Z [2] to each small $\mathrm{Z_i}$, while developing a novel cross-view alignment regularization to couple multi-view data to take full advantage of label information. Another thing is we can only achieve local minimal solutions for both versions. And we have more variables to be optimized in this journal version so that we may not achieve more optimal solutions than our conference version [2] when $\lambda_2 = 0$. We will show this phenomenon in the experiments.

2.2.7 Complexity Analysis

In this section, we provide a detail complexity analysis of our algorithm. Suppose $\mathrm{X}_i \in \mathbb{R}^{d\times m}$ and $\mathrm{D} \in \mathbb{R}^{d\times \bar{m}}$, where d is the original dimensionality of data, m is the size of the i-th view data X_i and $\bar{m}$ is the size of view-mixed data D. And we assume P_i and P are all $d \times p$ matrices, where p is the reduced dimensionality of subspace. The major time-consuming parts of **Algorithm 1** are: 1) SVD operation in Step 1-1; 2) Matrix inverse and multiplication in Step 1-2 and 1-4; 3) Subspace Optimization in Step 2; 4) Sylvester equation in Step 3.

We now discuss each part in detail. Since $k \ll m$, $\mathcal{O}(m) \approx \mathcal{O}(\bar{m})$. For simplicity, J_i, Q_i, Z can be treated as $m \times m$ matrices. For P_i and P are $p \times d$ matrices. First of all, nuclear norm in Step 1-1 costs $\mathcal{O}(m^3)$ through SVD operation. Fortunately, according to Theorem 4.3 [50], the SVD for Z_i could be speeded up to $\mathcal{O}(p^2 m)$ where p is usually a small one. Secondly, we calculate the matrix multiplication and inverse. Step 1-2 would cost about $\mathcal{O}(m^3)$, while Step 1-4 will each approximately take $\mathcal{O}(d^3)$. Step 2 would cost $\mathcal{O}(\tau d^3)$ if there are τ iterations within Step 2. Finally, Step 3 generally takes $\mathcal{O}(m^3)$ to optimize $\mathrm{Z} \in \mathbb{R}^{m\times \bar{m}}$ in Sylvester function. To sum up, we conclude that the time complexity of CLRS is $\mathcal{O}(d^3 + m^3)$.

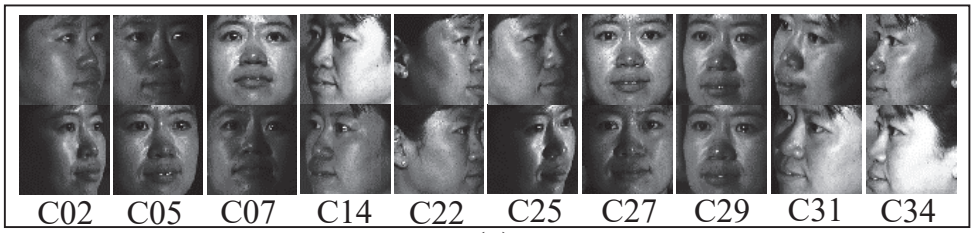


Figure 2.2: Face samples from different views of one individual in CMU-PIE cross-pose face dataset. It can be observed that the dissimilarity across different views of the same individual.

Table 2.1: Recognition Performance (%) of 10 algorithms on the **original images** from CMU-PIE face dataset, in which Case 1: {C02, C14}, Case 2: {C02, C27}, Case 3: {C14, C27}, Case 4: {C05, C07, C29}, Case 5: {C05, C14, C29, C34}, Case 6: {C02, C05, C14, C29, C31}

	PCA[64]	LDA[65]	LPP[66]	RSR[67]	TFRR[68]	SRRS[51]	LRCS [2]	MvDA[1]	RMSL[39]	Ours
Case 1	69.03±0.08	70.46±0.05	57.25±0.06	77.51±0.01	77.92±0.03	78.27±0.04	87.78±0.22	85.23±0.05	**88.15±0.06**	87.24±0.03
Case 2	69.21±0.08	71.32±0.02	58.83±0.07	74.74±0.17	76.24±0.12	78.74±0.23	86.67±0.09	85.81±0.09	**87.05±0.07**	86.82±0.11
Case 3	68.52±0.12	63.51±0.75	59.25±0.56	71.10±0.04	75.29±0.07	77.45±0.02	87.38±0.39	86.12±0.12	87.40±0.17	**87.97±0.09**
Case 4	52.65±0.04	56.53±0.02	43.56±0.08	67.57±0.01	69.74±0.05	71.44±0.03	74.84±0.04	**75.36±0.18**	75.16±0.12	72.97±0.03
Case 5	34.94±0.08	24.07±0.25	19.67±0.05	29.72±0.01	33.91±0.12	38.86±0.02	44.48±0.03	**54.13±0.16**	44.93±0.11	45.92±0.06
Case 6	29.09±0.01	07.06±0.01	13.11±0.01	09.44±0.02	28.36±0.04	30.16±0.02	36.17±0.11	**47.67±0.18**	37.14±0.08	39.17±0.08

2.3 Experiment

In this section, we first introduce the real multi-view datasets (e.g., cross-pose and cross-modality data) and experimental settings. Then, we compare with the state-of-the-art algorithms in two different scenarios. Finally, we evaluate some properties of our proposed CLRS.

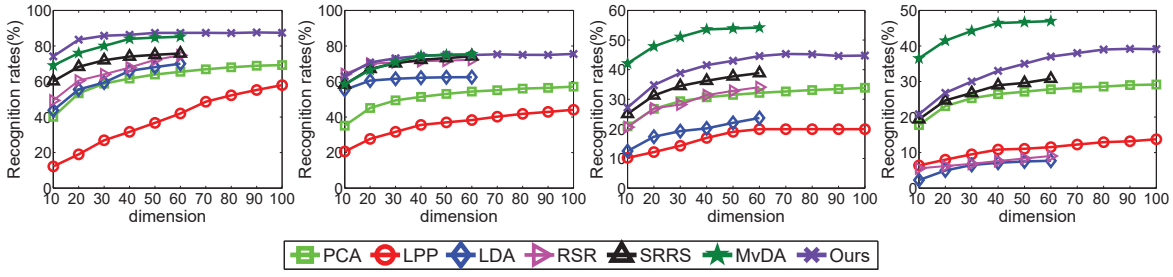


Figure 2.3: Recognition performance of 7 algorithms on the **original images** of CMU-PIE face dataset over different dimensions, which shows the performance of Case 2, Case 4, Case 5 and case 6, from left to right. We can only obtain at most 67 dimensions for the LDA-based algorithms (LDA, RSR, SRRS and MvDA) (Here we only present 60 dimensions for them).

Table 2.2: Recognition Performance (%) of 10 algorithms on the **corrupted images** from CMU-PIE face dataset, in which Case 1: {C02, C14}, Case 2: {C02, C27}, Case 3: {C14, C27}, Case 4: {C05, C07, C29}, Case 5: {C05, C14, C29, C34}, Case 6: {C02, C05, C14, C29, C31}

	PCA[64]	LDA[65]	LPP[66]	RSR[67]	TFRR[68]	SRRS[51]	LRCS [2]	MvDA[1]	RMSL[39]	Ours
Case 1	64.87±0.32	26.71±0.20	31.26±0.26	37.02±0.03	68.10±0.07	72.27±0.05	78.98±0.03	75.34±0.09	**81.12±0.08**	80.58±0.05
Case 2	66.04±0.08	23.19±0.35	30.98±0.18	34.34±0.15	68.24±0.32	72.74±0.18	78.67±0.05	74.81±0.09	**81.67±0.09**	81.06±0.06
Case 3	65.21±0.04	20.34±0.75	32.21±0.36	31.69±0.09	67.85±0.12	71.45±0.08	78.38±0.26	76.24±0.15	81.08±0.17	**82.11±0.18**
Case 4	50.16±0.04	16.72±0.02	27.66±0.05	22.45±0.01	50.94±0.09	54.32±0.03	65.84±0.04	61.26±0.12	66.95±0.08	**67.10±0.04**
Case 5	31.74±0.08	06.67±0.25	14.34±0.04	10.02±0.01	29.26±0.12	32.34±0.02	39.48±0.03	**44.19±0.13**	43.87±0.11	41.68±0.03
Case 6	27.21±0.01	04.06±0.01	12.02±0.01	04.95±0.02	28.12±0.03	29.03±0.02	32.57±0.01	**34.17±0.21**	33.78±0.08	33.67±0.08

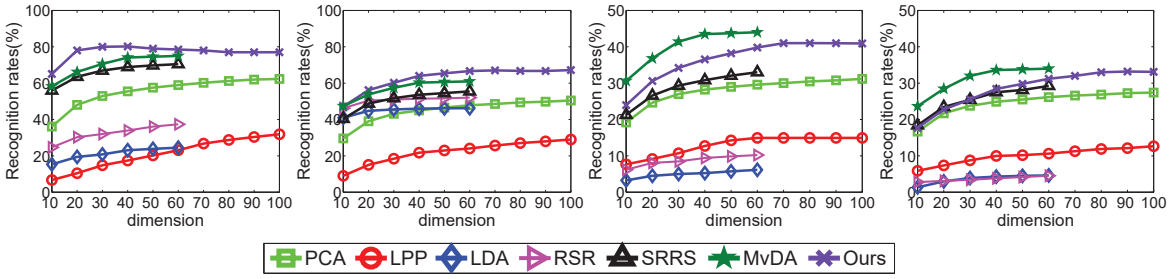


Figure 2.4: Recognition performance of 7 algorithms on the **corrupted images** of CMU-PIE face dataset over different dimensions, which shows the performance of Case 2, Case 4, Case 5 and case 6, from left to right. We can only obtain at most 67 dimensions for the LDA-based algorithms (LDA, RSR, SRRS and MvDA) (Here we only present 60 dimensions for them).

2.3.1 Datasets & Experimental Setting

CMU-PIE Face dataset [69] totally consists of 68 subjects with different poses. There is 21 different illumination variations for the samples of each subject. Specifically, we adopt such different poses, which show large view variances within the same subject across different poses (Fig. 3.3 (a)). In the experiment, we select different numbers of views to build various evaluation scenarios. For each pose, we randomly choose 10 samples for training while the left for testing. Furthermore, we crop faces into size of 64×64 and adopt the gray-scale value as the input.

BUAA VIS-NIR face dataset [70] contains 150 different individuals, and every individuals has two modalities, i.e., near-infrared faces (NIR) & visible faces (VIS). Specifically, there are 9 face images per modality per individuals. We further crop the faces and resize them to the size 200×200. The gray-scale value is used as the input feature.

2.3.2 Feature Representation Setting

In our experiment, we address the challenging problem where the view knowledge of the probe data is unavailable. Thus conventional multi-view methods [1, 37] would fail. Therefore, we mainly compare with PCA [64], LDA [65], LPP [66], RSR [67], TFRR [68], SRRS [51], RMSL [39] and LRCS [2]. Specifically, LDA, RSR, SRRS, RMSL and ours are five supervised algorithms; and PCA, LPP, TFRR and LRCS are four unsupervised methods. Furthermore, we compare with one conventional multi-view subspace learning algorithm, MvDA [1], by providing it extra view knowledge of the probe data to show the effectiveness of our algorithm.

In this setting, we conduct experiment on the CMU-PIE face dataset. The nearest neighbor classifier (NNC) is adopted to testify the final classification results. For CMU-PIE, we choose ten images per individual per pose to build the training set, and the remaining data are used for testing. We do 5 random selections and report the average performance. Table 2.1 and Table 2.2 represent recognition performance on the original images and 10% corrupted images, respectively. Besides, we also evaluate their recognition performance under different dimensions in Figs. 3.6, 3.7.

Discussion: Results demonstrate our method outperforms others in most cases, except MvDA and RMSL. However, we can see our algorithm could achieve competitive performance with MvDA, or even better in some cases. This demonstrates our algorithm is an effective compromise when we are inaccessible to the view information of the probe data. However, when there are more views, MvDA has a superiority in performance, since multiple view-specific transformations could well fit each specific view data. Besides, MvDA is one kind of traditional subspace learning methods, which cannot work well in corrupted cases even though the view knowledge of the probe data is available. Our algorithm unifies low-rank reconstruction and dimension reduction together, and therefore it could well handle the corrupted data in reality.

With more poses involved, we could observe that all the algorithms suffer a decrease in terms of recognition performance. However, we notice that the performance of LDA and RSR decrease much faster, which shows that conventional subspace learning algorithms, e.g., LDA, would fail in multi-view learning. For 2-view scenarios (i.e., Case 1-3), all the methods can obtain very similar results, which represents that the divergence across any two views is very similar. As we can see, these three cases effectively show the superiority of our model. This denotes that the collective low-rank common subspace, decomposed from two view-specific transformations, can uncover the most intrinsic information from the data. Moreover, for 3-view case (Case 4), our model cannot improve with a large margin, and the reason we consider is that three views are relatively frontal

Table 2.3: Recognition performance (%) 10 algorithms on **Group 1** of CMU-PIE dataset, in which Case 1: {C02, C14}, Case 2: {C02,C27}, Case 3:{C14,C27}, Case 4: {C05, C07, C29}, Case 5: {C05, C14, C29, C34}, Case 6: {C02, C05, C14, C29, C31}

Methods	PCA [64]	LDA[65]	LPP[66]	TSL [71]	GFK [9]	LTSL [16]	DASA [12]	JDA [72]	LRCS[2]	Ours
Case 1	34.87±0.32	23.71±0.20	29.26±0.26	49.43±0.09	56.21±0.03	58.29±0.01	57.34±0.03	60.91±0.08	61.23±0.12	**62.12±0.09**
Case 2	36.04±0.08	21.19±0.35	28.98±0.18	48.88±0.12	55.48±0.08	57.68±0.06	56.84±0.06	60.18±0.12	60.62±0.10	**61.32±0.09**
Case 3	34.21±0.04	19.84±0.75	30.21±0.36	49.72±0.08	56.12±0.08	58.36±0.08	57.42±0.10	61.23±0.11	61.37±0.09	**61.92±0.13**
Case 4	23.16±0.04	16.72±0.02	21.66±0.05	29.28±0.05	34.87±0.05	37.66±0.06	35.62±0.12	48.59±0.08	49.09±0.09	**49.92±0.13**
Case 5	13.74±0.08	06.67±0.25	12.34±0.04	19.34±0.08	23.29±0.08	28.11±0.05	26.23±0.09	32.86±0.08	33.05±0.04	**34.16±0.08**
Case 6	10.21±0.01	04.06±0.01	10.02±0.01	16.76±0.14	19.45±0.12	22.54±0.03	20.45±0.09	26.05±0.15	28.05±0.05	**30.03±0.09**

Table 2.4: Recognition performance (%) of 10 algorithms on **Group 2** of BUAA NIR-VIS face dataset.

Methods	PCA [64]	LDA[65]	LPP[66]	TSL [71]	GFK [9]	LTSL [16]	DASA [12]	JDA [72]	LRCS [2]	Ours
Case 1	46.23±0.21	45.83±0.27	48.07±0.18	56.02±0.42	65.82±0.26	68.07±0.15	62.12±0.32	68.01±0.21	70.56±0.20	**71.23±0.32**
Case 2	47.23±0.32	46.62±0.34	49.41±0.17	57.32±0.24	67.56±0.30	69.17±0.19	63.45±0.45	68.93±0.12	71.63±0.22	**72.22±0.29**

faces. When involving more views, (e.g., Case 5 and 6), our proposed algorithm can still perform better than others.

Another observation is that low-rank based algorithms (i.e., TFRR, SRRS, LRCS, RMLS and ours) can work better than the other algorithms, especially when dealing with corrupted cases. The reason we consider is that low-rank based algorithms are able to capture the data's intrinsic class-wise structure. Another phenomenon is each pose show 21 different lighting variations in CMU-PIE face dataset, therefore, PCA has the similar results in the original & randomly corrupted images.

Compared with our previous conference version, our CLRS works a little worse in two cases. The reason, we consider, is the solution to the objective function since our current one is supervised. We adopt multi-task learning technique to solve multiple view-specific transformations in parallel, which could make the optimal solutions more flexible, compared with our conference solution. This phenomena is discussed more in the following parameter analysis section.

2.3.3 Transfer Learning Setting

In this part, experiments are conducted to evaluate the algorithms in transfer learning setting. We have 6 comparisons, i.e., JDA [72], LTSL [16], GFK [9], DASA [12], TSL [71] and LRCS [2]. Also we compare with some conventional subspace learning methods, e.g., PCA [64],

LDA [65], LPP [66]. For those methods, we apply them on source and target data together (except LDA, which is supervised, so only source data is used) to learn the projection in the training stage, then predict the unlabeled target data. Finally, the nearest neighbor classifier (NNC) is adopted to testify the effectiveness for target domain.

Group 1: To construct two domains, we first split CMU-PIE with 68 individuals into 2 subsets, each with 34 different individuals. To make source and target with different distributions, we utilize low-resolution process for the source domain. In target domain, we adopt the original 64×64 images. In source domain, the 64×64 images (HR) are resized to 16×16, and resized back to the original size (LR). The Matlab function *imresize()* with default setting is used. We choose different poses in the same previous setting to "feature representation setting". Since there is no label overlap across two domains, therefore, we randomly choose 2 reference face images per view in the target domain in the evaluation stage, while other samples are used for evaluation. Ten random selections are conducted and the average results are reported.

Group 2: we split the BUAA NIR-VIS into two subsets, one as source and the other as target. In this dataset, we conduct two different cases. Case 1: choosing 50 individuals as source, the left 100 individuals as target; Case 2: choosing 75 individuals as source, the left 75 individuals as target. Every individual contains two modalities. Note that no identity overlap exits across source and target. To further differentiate source and target, we exploit down-sampling procession to source. Finally, we randomly select 2 target samples per individual as the reference, while the left target samples are used for evaluation. We randomly select for 9 times, then calculate the average results.

Discussion: It can be seen from Table 2.3, and 2.4 that our proposed model achieves better performance than others. Since we seek multiple view-specific transformations on the well-labeled source domain, our proposed algorithm could effectively transfer such multi-view knowledge to the unlabeled target domain by coupling various views properly with the collective low-rank projection. Moreover, we could observe that low-rank based algorithms, i.e., LTSL, LRCS and ours, outperform the other algorithms. Furthermore, we could notice that the effectiveness of CLRS is significant in Group 1, but not obvious in Group 2. The reason may be the domain divergence is larger in Group 1.

2.3.4 Property Evaluation

In this section, we evaluate several properties of our proposed algorithm, i.e., convergence analysis, parameter sensitivity and training time cost.

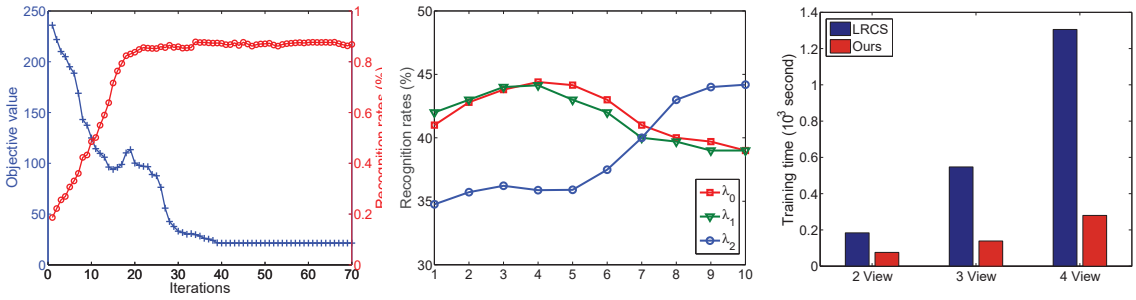


Figure 2.5: **Left**: Convergence curve (Blue '+') and recognition curve (red 'o') of CLRS for Case 2 {C02,C27} of CMU-PIE face dataset, where $p = 100, \lambda_0 = \lambda_1 = 10^{-2}, \lambda_2 = 10^2$. Here we represent the results of 70 iterations. **Middle:** Influence of parameters $\lambda_0, \lambda_1, \lambda_2$ on the case of {C05, C14, C29, C34}. The value from 1 to 10 represents $[0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4]$. **Right**: Training cost (*second*) of two methods on CMU-PIE face dataset.

2.3.4.1 Convergence Analysis

In this part, we mainly verify the influence of the parameters and some properties of convergence. We evaluate on robust feature learning setting and use CMU-PIE database with Case 2 {C02,C27}.

We analyze the convergence through experiments with different runs. The convergence curve of our algorithm is shown in Fig. 2.8 (Left), together with the recognition results with increasing iterations. From the results, we can observe our method converges after 50 iterations. Another phenomenon is the recognition results of our CLRS would reach the highest after about 30 iterations and keep almost stable. Therefore, we usually choose 50 iterations to optimize the collective subspace P to achieve the final results.

2.3.4.2 Parameter Analysis

Moreover, we have three parameters $\lambda_0, \lambda_1, \lambda_2$. In this section, we mainly evaluate the balanced parameter $\lambda_0, \lambda_1, \lambda_2$ on one 4-view case (Fig. 2.8 (Middle)). From the results, we can observe the large value of λ_2 performs better, that means this term can effect more. While our model achieves better performance when λ_0 and λ_1 are with small values. Without loss of generality, we set $\lambda_0 = \lambda_1 = 10^{-2}$ and $\lambda_2 = 10^2$ in our experiments.

Remark: When λ_2 becomes zero, the proposed method degenerates to our conference version, i.e., LRCS [2]. However, we apply multi-task learning technique to solve the problem in the journal extension to speed up the optimization, which is different from our conference version. When we

adopt multi-task learning technique, multiple view-specific projections are optimized in an individual way, which results in more variables needed to be updated. Since we can only achieve local minimum solutions through ADMM, more variables may produce more flexible optimization, therefore the proposed algorithm may not find better optimal solutions than our conference version, where multiple view-specific projections are stacked together and optimized as a whole. These are the cases when the proposed method fails to outperform our conference version, i.e., $\lambda_2 = 0$.

2.3.4.3 Training Time Analysis

We evaluate the computational cost of our previous work [2] and our current version. We conduct on various cases on CMU-PIE dataset and report the training cost with 10 iterations. We evaluate on Matlab 2014 with CPU i7-3770 while 32GB memory size. Fig. 2.8 (Right) shows the training time, whose unit is *second*.

From Fig. 2.8 (Right), we notice that the proposed algorithm performs more efficiently than our previous work, as we mentioned before. The most time-consuming part is the low-rank term on the collective projection P for the high dimension. When more views involved, our previous work costs more time than our current version. We attribute to the efficient multi-task scheme to the optimization solution.

2.4 Deep Multi-View Face Recognition

To explore deep learning, we propose a Deep Generalized Adaptive Network framework (DGAN) by leveraging the knowledge between multiple source views and the unseen target views (Fig. 2.6), where (a) Multiple view-specific deep structures $\{W_i^{(l)}, b_i^{(l)}, i = 1, \cdots, \mathrm{M}\}$ tend to be learned to capture the rich information from each source; (b) A view-invariant deep structure $\{W_c^{(l)}, b_c^{(l)}\}$ is built for all the views, and further generalize to the view-unseen data in the testing stage; (c) To couple the outputs of multiple view-specific networks $\{\mathbf{H}_1^{(L)}, \cdots, \mathbf{H}_{\mathrm{M}}^{(L)}\}$ and view-invariant one $\mathbf{H}_c^{(L)}$, low-rank reconstruction is adopted to align two types of networks in class-wise fashion. The core idea of DGAN is to seek a view-invariant deep structure by uncovering shared knowledge across multiple source views to generalize to unseen target view. The main contributions are summarized as follows:

- Multiple view-specific deep structures and one view-invariant deep structure are jointly learned to uncover more useful information from each view and shared by different views, respectively.

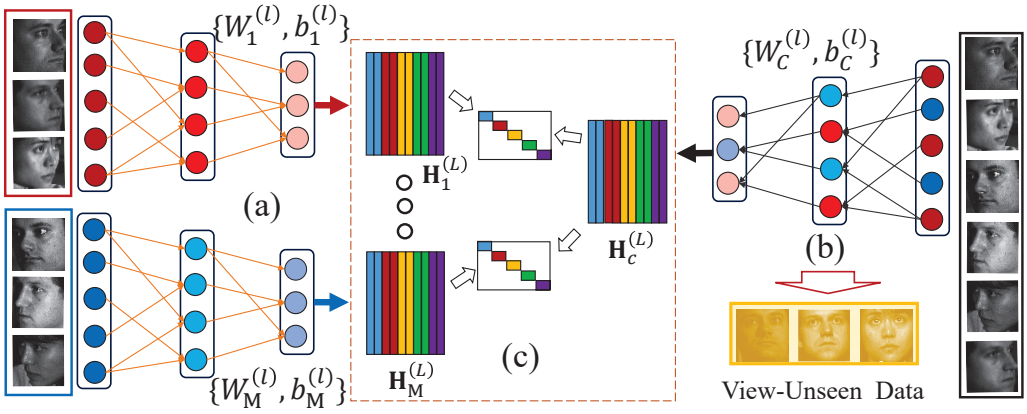


Figure 2.6: Framework of our proposed algorithm.

With multi-layer networks, the rich knowledge within sources can be learned to facilitate the view-unseen face data learning.

- To better couple multiple view-specific structures and the view-invariant one, we deploy a class-wise low-rank reconstruction scheme to mitigate the knowledge between view-specific and the view-invariant one. Specifically, the output from each view-specific network would be only reconstructed by the output from the view-invariant network with the same class label under low-rank constraint. To this end, the learned view-invariant deep structure can be applied to the view-unseen faces.

2.4.1 The Proposed Algorithm

In this section, we will introduce our proposed Deep Generalized Adaptive Network for unseen target views by learning multiple view-specific and one view-invariant deep neural networks from several related source views. Then, we provide the solution to the proposed algorithm.

In reality, multi-view face data are very common and popular, since different views could facilitate the face data representation, for example, near-infrared face images are more insensitive to the illuminations. However, multi-view face data within the same class could show definitely large diversity, resulting in a challenging issue in multi-view learning. Take cross-pose face recognition [37] for example, we can observe that the pose variance within the subject could be extremely large so that the similarity of the same class samples in different poses would be lower than that of same pose samples from different classes. To this end, face data from the same view across different

classes tend to lie in one view-specific subspace. This phenomena of multi-view face data would definitely degrade the classification performance.

Fortunately, data collected from various views share the same class information, therefore, each view-specific projection should have similar discriminability to separate different classes for individual view. Kan et al. mentioned that the structure of each projection for multi-view data is similar, that is view-consistency [52]. In this way, it is reasonable to consider that the multiple view-specific nonlinear projection should share a lot of consistent information [22]. To this end, we design two types of deep structures, i.e., view-specific and view-invariant, to facilitate the view-unseen face learning.

2.4.1.1 Deep Generalized Adaptive Network

Deep neural networks aim to seek a compact representation for each sample $x \in \mathbb{R}^{d_0}$ by passing it through stacked multiple layers of nonlinear transformations. The major merit of such networks is that the nonlinear mapping function can be explicitly obtained for better feature extraction. Assume there are $L+1$ layers in the designed network and d_l units in the l-th layer, where $l = 1, \cdots, L$. The output of x at the l-th layer is computed as:

$$f^{(l)}(x) = \mathbf{h}^{(l)} = \varphi(W^{(l)}\mathbf{h}^{(l-1)} + b^{(l)}), \tag{2.19}$$

where $W^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{d_l}$ are the weight matrix and bias of the parameters in l-th layer; $\mathbf{h}^{(l)}$ is the l-th hidden layer and $\mathbf{h}^{(0)} = x$; φ is a nonlinear activation function which operates component-wise. The overall nonlinear mapping $f^{(L)} : \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_L}$ is a function parameterized by $\{W^{(l)}\}_{l=1}^{L}$ and $\{b^{(l)}\}_{l=1}^{L}$.

In multi-view learning, assume we have M source views data $\{(\bar{X}, \bar{y})\} = \{(X_1, y_1), \cdots, (X_{\mathrm{M}}, y_{\mathrm{M}})\}$, where $X_i \in \mathbb{R}^{d_1 \times n_i}$ is the i-th view with n_i samples of dimension d and y_i is the label vector of i-th view. We design M view-specific deep structures for each source view and one view-invariant deep structure for all the views, including the unseen target views. Specifically, we have the i-th view-specific deep network as:

$$f_i^{(l)}(X_{i,j}) = \mathbf{H}_{i,j}^{(l)} = \varphi(W_i^{(l)}\mathbf{H}_{i,j}^{(l-1)} + b_i^{(l)}), \tag{2.20}$$

where $W_i^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_i^{(l)} \in \mathbb{R}^{d_l}$ are the weight matrix and bias of the parameters in l-th layer. $X_{i,j}$ is the j-th sample in the i-th source view and $\mathbf{H}_{i,j}^{(l)}$ is the l-th hidden layer of $X_{i,j}$. The view-invariant network is designed as follows:

$$f_c^{(l)}(\bar{X}_j) = \mathbf{H}_{c,j}^{(l)} = \varphi(W_c^{(l)}\mathbf{H}_{c,j}^{(l-1)} + b_c^{(l)}), \tag{2.21}$$

where $W_c^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_c^{(l)} \in \mathbb{R}^{d_l}$ are the weight matrix and bias of the parameters in l-th layer. $\bar{X}_j$ is the j-th sample in combined source views $\bar{X}$ and $\mathbf{H}_{c,j}^{(l)}$ is the l-th hidden layer of $\bar{X}_j$. Currently, view-specific deep networks and view-invariant deep network are learned separately. Therefore, knowledge across multiple source views cannot be transferred to each other, let alone the view-invariant one, which is designed to extend to the unseen target views.

To guide two types of networks learning and transfer more effective knowledge to the view-unseen face evaluation, we propose to reconstruct the output of each view-specific network with the output of the view-invariant network under low-rank constraints as follows:

$$\min_{Z^i} \sum_{i=1}^{\mathrm{M}} \mathrm{rank}(Z^i), \ \text{s.t.} \ \mathbf{H}_i^{(L)} = \mathbf{H}_c^{(L)} Z^i, \tag{2.22}$$

where $\mathbf{H}_i^{(L)} \in \mathbb{R}^{d_L \times n_i}$ is the output of the i-th view-specific network, while $\mathbf{H}_c^{(L)} \in \mathbb{R}^{d_L \times n}$ is the output of the view-invariant network ($n = \sum_i n_i$). $Z^i \in \mathbb{R}^{n \times n_i}$ is the reconstruction coefficient matrix for each view and $\mathrm{rank}(\cdot)$ is the rank operator on a matrix. The low-rank reconstruction coefficient matrix is block-diagonal in the ideal case, that is, only the data with same class label are correlated between two types of deep networks. Similar ideas were proposed in the literature [2] which adopted multi-view data to reconstruct each view under linear transformations with low-rank constraint. However, such shallow structures with linear transformations would fail to uncover the rich and complex information within data.

To solve problem (2.22), we propose to construct a cross-network weight matrix that describes the locality-aware similarities between two types of deep networks. To avoid the complex rank constraint, we could solve the reconstruction problem in a sample-wise way, meaning only one sample is reconstructed each time. To find the best reconstruction coefficients, we can use least square loss criterion with l_1 (lasso) or l_2 (ridge regression) regularizer, and enforce the larger coefficients only from the neighborhood of the input. However, this may still take a long time when the scale of each domain is large. We further reduce the time complexity for building each low-rank reconstruction coefficient matrix Z_i through Nadaraya-Watson Kernel as:

$$Z_{jk}^i = \frac{\mathcal{K}(X_{i,j}, \bar{X}_k)}{\sum_{y_{i,j}=\bar{y}_k} \mathcal{K}(X_{i,j}, \bar{X}_k)}, \ i = 1, \cdots, \mathrm{M}, \tag{2.23}$$

where $\mathcal{K}(X_{i,j}, \bar{X}_k) = \exp(-\|X_{i,j} - \bar{X}_k\|^2 / 2\sigma^2)$ is Gaussian kernel function with σ as bandwidth (we set $\sigma = 5$ in our experiment). Clearly, using Eq. (2.23) to construct the low-rank coefficients is much faster than previous linear coding or rank constraint based method. In addition, it is a locality-aware weight matrix, since it is built based on within-class samples for each sample. To this

end, we propose our class-wise low-rank reconstruction as:

$$\Omega_z = \sum_{i=1}^{\mathrm{M}} \|\mathbf{H}_i^{(L)} - \mathbf{H}_c^{(L)} Z^i\|_{\mathrm{F}}^2, \tag{2.24}$$

where each Z^i is pre-learned from Eq. (2.23).

Since multiple source views all share the same categories but in different distributions, we assume the view-specific deep networks share most information in the latent space, where the knowledge can be extended to the unseen target view [22]. An intuitive strategy is to couple the weights in each layer between the view-specific networks and the view-invariant one. To this end, we build the following connections across them as:

$$\Omega_c = \sum_{i=1}^{\mathrm{M}} \sum_{l=1}^{L} (\|W_i^{(l)} - W_c^{(l)}\|_{\mathrm{F}}^2 + \|b_i^{(l)} - b_c^{(l)}\|_2^2). \tag{2.25}$$

In this way, the common view-invariant network can uncover shared factors across multiple view-specific networks so that it can be better extended to alleviate the unseen target view learning in the real testing stage.

To further make full use of the label information, we adopt two supervised regularizers to guide the learning of the view-invariant networks. We define the intra-class compactness as Ω_w and the inter-class separability Ω_b as:

$$\begin{aligned} \Omega_w &= \sum_{k=1}^{n} \sum_{j=1}^{n} W_{kj}^w \|\mathbf{H}_{c,k}^{(L)} - \mathbf{H}_{c,j}^{(L)}\|_2^2, \\ \Omega_b &= \sum_{k=1}^{n} \sum_{j=1}^{n} W_{kj}^b \|\mathbf{H}_{c,k}^{(L)} - \mathbf{H}_{c,j}^{(L)}\|_2^2, \end{aligned} \tag{2.26}$$

where $\mathbf{H}_{c,k}^{(L)}$, $\mathbf{H}_{c,j}^{(L)}$ are the k-th and j-th columns of $\mathbf{H}_c^{(L)}$. W_{kj}^w is set as one if $\bar{X}_j$ is one of k_1 intra-class nearest neighbors of $\bar{X}_k$, and zero otherwise; and W_{kj}^b is set as one if $\bar{X}_j$ is one of k_2 inter-class nearest neighbors of $\bar{X}_k$, and zero otherwise. Similar strategy has been adopted in [17].

To sum up, we develop our deep generalized adaptive networks framework by minimizing the objective function:

$$\begin{aligned} J &= \Omega_z + \alpha\Omega_w - \beta\Omega_b + \lambda\Omega_c, \\ &= \sum_{i=1}^{\mathrm{M}} \|\mathbf{H}_i^{(L)} - \mathbf{H}_c^{(L)} Z^i\|_{\mathrm{F}}^2 + \mathrm{tr}(\mathbf{H}_c^{(L)} \mathcal{L} (\mathbf{H}_c^{(L)})^\top) + \lambda \mathcal{N}_c, \end{aligned} \tag{2.27}$$

where $\alpha > 0$, $\beta > 0$ and $\lambda > 0$ are three positive trade-off. $\mathcal{L} = \alpha L_w - \beta L_b$ and L_w, L_b are the Laplacian graph of the intra-class and inter-class matrix, respectively. $\mathrm{tr}(\cdot)$ is the trace operator

of a matrix. With Eq. (2.27), two types of networks are well coupled under class-wise low-rank constraints in order to transfer more knowledge from view-specific networks to the view-invariant one. Furthermore, the intra-class and inter-class structures tend to make the view-invariant network more discriminative.

Discussion: Our proposed algorithm builds two types of deep structures to extract most shared information across multiple related source views and transfer the knowledge to the unseen target views in the testing stage. The most correlated work is MTAE [27], which exploited the common encoder to transform multiple sources to the hidden layer, then adopted domain-specific decoders. In this way, the common domain-invariant encoder could be generalized to unseen target domains. The connection in MTAE between the domain-specific and the domain-invariant part is the same sample (point-to-point scheme), that is, MTAE ignored the intra-class/inter-class information in the encoding and decoding steps, where MTAE worked in an unsupervised manner. However, we adopt class-wise low-rank reconstruction to adapt the two types of deep networks, so that the same class data in two types of networks are correlated to mitigate the within-class variance. Furthermore, much deeper structures would also contribute to the discriminative feature learning across multiple sources and better facilitate the view-unseen target data learning.

2.4.1.2 Model Training

To solve the optimization problem in (2.27), we employ the stochastic sub-gradient descent method to obtain the parameters $W_i^{(l)}$, $b_i^{(l)}$, $W_c^{(l)}$ and $b_c^{(l)}$. The gradients of the objective function J in (2.27) with respect to the parameters $W_i^{(l)}$, $b_i^{(l)}$, $W_c^{(l)}$ and $b_c^{(l)}$ are computed as follows:

$$\frac{\partial J}{\partial W_i^{(l)}} = L_i^{(l)} \mathbf{H}_i^{(l-1)^\top} + 2\lambda(W_i^{(l)} - W_c^{(l)}), \tag{2.28}$$

$$\frac{\partial J}{\partial b_i^{(l)}} = \overline{L}_i^{(l)} + 2\lambda(b_i^{(l)} - b_c^{(l)}), \tag{2.29}$$

$$\frac{\partial J}{\partial W_c^{(l)}} = L_c^{(l)} \mathbf{H}_c^{(l-1)^\top} + 2\sum_{i=1}^{\mathrm{M}} \lambda(W_c^{(l)} - W_i^{(l)}), \tag{2.30}$$

$$\frac{\partial J}{\partial b_c^{(l)}} = \overline{L}_c^{(l)} + 2\sum_{i=1}^{\mathrm{M}} \lambda(b_c^{(l)} - b_i^{(l)}), \tag{2.31}$$

where the updating equations are computed as follows:

$$L_i^{(L)} = 2(\mathbf{H}_i^{(L)} - \mathbf{H}_c^{(L)} Z_i) \odot \varphi'(U_i^{(L)}),$$

$$L_c^{(L)} = 2\Big(\sum_{i=1}^{\mathrm{M}} (\mathbf{H}_i^{(L)} - \mathbf{H}_c^{(L)} Z_i) Z_i^\top + \mathcal{L}\mathbf{H}_c^{(L)^\top} \Big) \odot \varphi'(U_c^{(L)}),$$

$$L_i^{(l)} = (W_i^{(l+1)^\top} L_i^{(l+1)}) \odot \varphi'(U_i^{(l)}),$$

$$L_c^{(l)} = (W_c^{(l+1)^\top} L_c^{(l+1)}) \odot \varphi'(U_c^{(l)}),$$

where $l = 1, 2, \cdots, L-1$. Here the operation $\odot$ denotes the element-wise multiplication, $U_i^{(l)} = W_i^{(l)} \mathbf{H}_i^{(l-1)} + \widetilde{b}_i^{(l)}$ and $U_c^{(l)} = W_c^{(l)} \mathbf{H}_c^{(l-1)} + \widetilde{b}_c^{(l)}$. $\overline{L}_c^{(l)}$ and $\overline{L}_i^{(l)}$ are the sum of all columns of $L_c^{(l)}$ and $L_i^{(l)}$, respectively. Whilst $\widetilde{b}_i^{(l)}$ and $\widetilde{b}_c^{(l)}$ are the n_i-time and n-time repeat of column of $b_i^{(l)}$ and $b_c^{(l)}$, respectively.

Then, $W_c^{(l)}$, $b_c^{(l)}$, $W_i^{(l)}$ and $b_i^{(l)}$ can be updated by using the gradient descent algorithm as follows until convergence:

$$W_{i/c}^{(l)} = W_{i/c}^{(l)} - \eta \frac{\partial J}{\partial W_{i/c}^{(l)}}, \;\; b_{i/c}^{(l)} = b_{i/c}^{(l)} - \eta \frac{\partial J}{\partial b_{i/c}^{(l)}}, \tag{2.32}$$

where η is the learning rate. The $\tanh(\cdot)$ function is adopted as the nonlinear activation function in our method. The initialized value of the learning rate η is set as 0.2, and then it gradually reduces by multiplying a factor 0.95 in each iteration. Other parameters are tuned in the experiments.

2.4.2 Experiments

In this section, we will evaluate our algorithm. First, we introduce the datasets and experimental settings. Then, we compare with several state-of-the-art methods to verify the superiority of the proposed algorithm. Finally, we analyze several properties of the proposed algorithm.

2.4.2.1 Datasets & Experimental Setting

Cross-pose Face CMU-PIE is combined by 68 subjects in total, which is a multi-pose face dataset [1]. Samples of each subject have 21 variations in lighting. We use five different poses (C05, C07, C09, C27, C29), which have large variances between the same subject at different poses. We crop images

[1] http://vasc.ri.cmu.edu/idb/html/face/

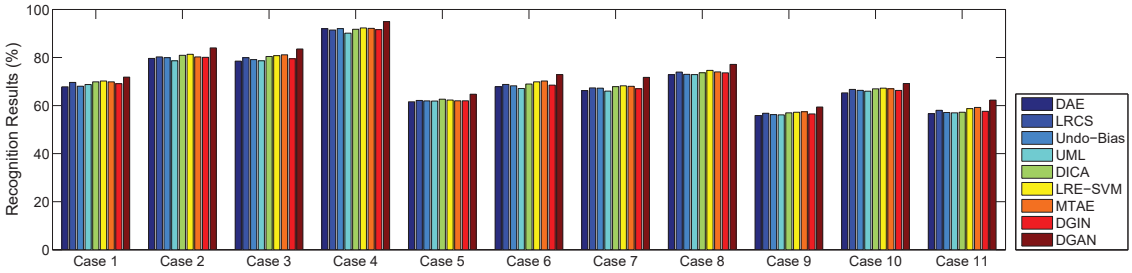


Figure 2.7: Recognition results on five cross-pose cases of CMU-PIE face database, where x-axis shows different combinations of view-unseen testing face and the value from Case 1 to Case 11 represents the unseen views are C05, C07, C09, C27, C29, {C05, C07}, {C05, C09}, {C05, C27}, {C05, C29}, {C09, C29}, {C05, C09, C29}, respectively. For each case, the left views out of five are used for training.

Table 2.5: Average Recognition Results (%) on YaleB face database, where Case 1: {R2, R3, R4} $\rightarrow$ {R1}, Case 2: {R1, R2, R3} $\rightarrow$ {R4}, Case 3: {R1, R2} $\rightarrow$ {R3, R4} .

Methods	DAE [30]	LRCS [2]	Undo-Bias [73]	UML [74]	DICA[75]	LRE-SVM [76]	MTAE [27]	DGIN	DGAN
Case 1	72.46	77.97	75.03	73.25	75.82	79.96	78.54	75.98	**81.23**
Case 2	65.17	74.54	69.56	72.33	74.97	80.15	79.65	73.24	**81.74**
Case 3	59.29	67.43	66.84	64.28	68.64	74.97	72.21	64.34	**76.72**
Average	65.64	73.31	70.04	69.95	73.14	78.36	76.80	71.18	**79.90**

into size of 64×64 and only use the raw images as the input. We build eleven different combinations to evaluate all the algorithms.

Yale B Face databases includes 38 subjects and the cropped images are used. The cropped original images are with size of 192×168, which we name it as R1. To differentiate images in different resolutions, we first downsample original images into $192/r \times 168/r$, then interpolate it back to 192×168. Hence we generate three more resolutions by setting $r = 4$ (R2), $r = 8$ (R3) and $r = 12$ (R4). Both of them are implemented by *imresize()* function in matlab. It can be observed that images of low-resolution are very blurry.

2.4.2.2 Comparison Experiments

In our experiment, we aim to address the challenging problem where the view information in testing face data is unknown. Thus the conventional multi-view learning algorithms [1, 37] would fail in this setting. Therefore, we mainly compare with DAE [30], LRCS [2], Undo-Bias [73], UML [74], LRE-SVM [76], MTAE [27] and DICA [75]. The last five algorithms are the state-of-the-art

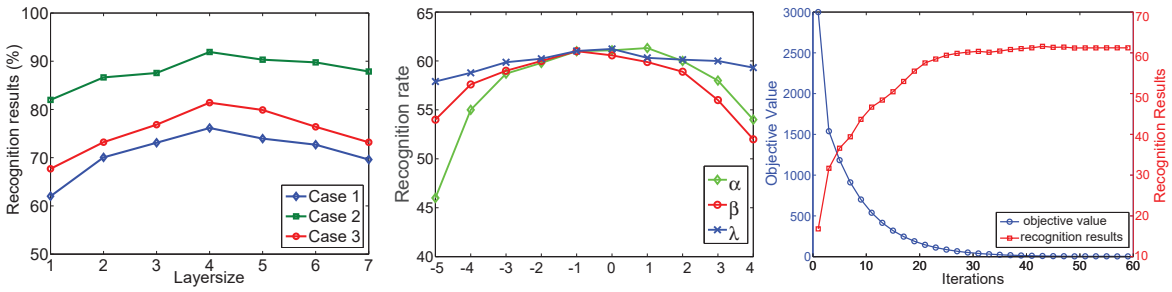


Figure 2.8: **Left**: Evaluation on different layer sizes from 1 to 7, where Case 1 means $\{C07, C09, C27\} \rightarrow \{C05, C29\}$, Case 2 denotes $\{C05, C07, C09, C29\} \rightarrow C27$ and Case 3 represents $\{$R2, R3, R4$\} \rightarrow \{$R1$\}$. **Middle**: Parameter analysis results on α, β, λ with the setting as $\{C07, C09, C27\} \rightarrow \{C05, C29\}$. **Right:** Convergence curve (blue) and recognition curve (red) of our algorithm on the setting $\{C07, C09, C27\} \rightarrow \{C05, C29\}$.

domain generalization methods, which tend to uncover more shared knowledge from multiple sources then generalize to unknown target. For our proposed algorithm, we both learn view-specific and view-invariant deep networks, and we adopt the common domain-invariant networks to extract features. With the extracted features, the nearest neighbor classifier (NNC) is adopted to apply the labeled source views to predict the labels of the view-unseen target faces. We adopt four-layer scheme to evaluate our algorithm and set $k_1 = 3, k_2 = 5$ for all the cases. To verify the effectiveness of our proposed model, we also evaluate the network using only the view-invariant deep structure with supervised term (Eq. (2.26)). We name it as DGIN and adopt the same layer-size to our DGAN. We report the performance in terms of the classification accuracy (%) [76]. For the algorithms that are optimized stochastically, we run independent training processes using the best performing hyper-parameters in ten times and report the average accuracies.

First of all, we observe that our proposed algorithm could outperform all the other comparisons in all the cases, since our proposed algorithm deploys two types of deeper structures and incorporates structured low-rank reconstruction fashion. Therefore, more discriminative knowledge could be transferred to facilitate the learning problem for view-unseen faces. Besides, domain generalization algorithms could achieve better results than DAE in most cases. That shows domain generalization techniques definitely facilitate the view-unseen face recognition by borrowing the knowledge from related multiple source views.

Secondly, from the results in Fig. 4.2, we observe that all the algorithms perform better when the testing face view is C27, that is, the frontal faces are much easier to recognize. All the other poses are near frontal faces, so traditional methods could achieve relative competitive

Table 2.6: Training time (*second*) on CMU-PIE face database.

Methods	Undo-Bias[73]	UML [74]	LRE-SVM [76]	DGAN
Cost	325.15	227.94	383.51	368.54

performance. However, most domain generalization algorithms can still boost the performance over DAE. Especially, our proposed algorithm can outperform all the competitors.

2.4.2.3 Empirical Analysis

First of all, we evaluate different layer sizes. To the best of our knowledge, there is no fixed rule to decide the layer size [22]. People usually empirically set the layer size based on the validation sets. Specifically, we compare the effect of shallow structure and deep structure for our proposed algorithm. The results in Fig. 2.8(Left) show that general four-layer structure would achieve better performance. With deeper structure, the performance may degrade somehow.

Secondly, we testify the influence of the three parameters α, β, λ in four-layer setting with dimensionality as [4096, 3000, 1000 200] on CMU-PIE cross-pose face database. Since there are three parameters, we evaluate one by fixing other two. The results are shown in Fig. 2.8(Middle), where **x**-axis values are $\log(x)$ processing. We can observe $\beta = 0.1$ performs better. We also observe $\lambda \in [0.1, 1]$ around 1 and $\alpha \in [0.1, 10]$ can achieve better performance.

Furthermore, we also evaluate the convergence and recognition results of our proposed algorithm with different iterations in optimization. We still use four-layer setting on $\{C07, C09, C27\} \rightarrow \{C05, C29\}$. The results are shown in Fig. 2.8(Right), where we can notice that our algorithm converges well, while the recognition results increase to the highest quickly and keep stable with more iterations.

Finally, we evaluate the model efficiency by running several algorithms on the CMU-PIE face database ($\{C05, C29\} \rightarrow C27$). Specifically,we report the average (training) runtime over all cross-domain recognition tasks in each dataset and show the results in Table 2.6. As we can see from the results, our proposed algorithm shows comparable time cost with some prior state-of-the-art domain generalization methods (i.e., Undo-Bias[73], UML [74] and LRE-SVM [76]).

2.4.3 Conclusion

In this paper, we developed a deep generalized adaptive network framework for view-unseen face evaluation, which aimed to seek most shared discriminative knowledge within multiple

source views to facilitate the viewunseen face learning. Specifically, we built two types of deep structures, view-specific and view-invariant, to capture most common discriminative information shared by multiple source views so that the knowledge could be transferred to the unseen target views. The class-wise low-rank constraints were adopted to mitigate the gap between view-specific and view-invariant structures. Furthermore, two supervised regularizations were proposed to fully utilize the label information. Experimental results demonstrated our proposed algorithm could outperform the state-of-the-art domain generalization methods.

Chapter 3

Transfer Learning for Face Recognition

3.1 Background

Transfer learning [77] attracts much interest in fields of machine learning and computer vision, as it is promising in handling the problem with sparse labeled data. Transfer learning usually borrows the knowledge from the existed well-labeled source data from other domains, subject to different feature spaces or distributions. In this line, recent work [47, 12, 78, 79, 80] has demonstrated that transfer learning can help achieve a good performance in this case. One solution to transfer learning is to discover a good feature representation of the data in one or two domains to mitigate the different marginal distribution or conditional distribution between two domains, which assumes that more sets of unlabeled target data are accessible for training the model [81, 12, 16, 82]. In reality, however, we always confront such a problem that no target data are achievable, especially when data are multi-modal [83, 84]. Under this situation, the target modality is blind in the training stage, while only the source modality can be obtained. We define such a problem as *Missing Modality Problem* in transfer learning.

Generally, in multi-modal transfer learning [85], knowledge is usually transferred from source to target modality assuming that both source and target modality are accessible in the training stage. In face recognition, such problem is quite common. To name a few: near-infrared (NIR) and visible light (VIS) images [83], sketches and photos [84], images in high resolution (HR) and low resolution (LR) [86]. The motivation for transfer learning between multi-modal data is clear: on one hand, we can easily achieve a lot of human-centered VIS images; on the other hand, we capture NIR images for identification due to its less sensitiveness to varied visible light. In such case, applying the well-learned knowledge from VIS images to help the recognition task of NIR images is non-trivial.

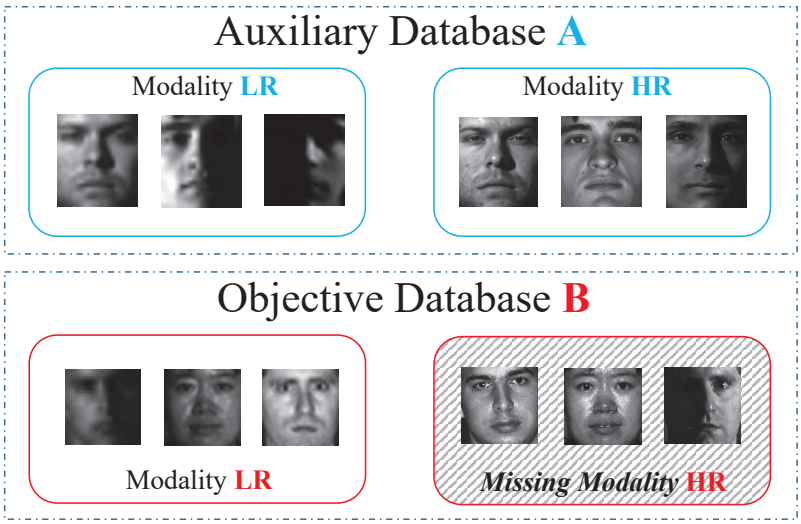


Figure 3.1: Ilustration of *Missing Modality Problem*. With the help of existed data (auxiliary database A and modality Low Resolution (LR) of B), the missing modality High Resolution (HR) of database B can be recovered to boost the recognition performance.

However, traditional transfer learning methods would fail if no training NIR images are available in the target domain. This is not rare because in many real-world systems, target data are only available at runtime. Due to significant difference between NIR and VIS images, direct use of VIS to recognize NIR images would yield an inferior performance.

Fortunately, we might be able to find similar multi-modal data from other databases with complete modalities. For example, we can transfer low resolution (LR) knowledge to the missing high resolution (HR) target if we have relevant yet slightly different HR and LR images from an auxiliary database (Fig. 5.4). In the ideal case, knowledge transferred between modality HR and LR from the auxiliary database A, shown in Fig. 5.4, can be immediately applied to recognize the missing modality in the objective database B, by assuming the transferred knowledge between HR and LR of databases A and B are identical. However, the transferred knowledge is not guaranteed to be same between database A and B since in reality lots of factors change from one database to another, e.g., capture environment, and devices. Therefore, knowledge transfer between two databases, especially when two databases are achieved under various conditions, needs extra care. In brief, although an auxiliary database casts a light on missing modality problem, it unfortunately brings in challenges as well.

From the analysis above, we can conclude that a straightforward approach to *Missing*

Modality Problem at least includes two steps: 1) transfer knowledge from auxiliary database to the objective one; 2) transfer knowledge from the source modality to the target one. Essentially, the conventional transfer learning process now is replaced by a transfer learning in two directions (Fig. 3.2). To the best of our knowledge, this is the first time that *Missing Modality Problem* is introduced under transfer learning framework.

Recently, low-rank matrix constraint [49, 87, 50] has been introduced into transfer learning problem. It is able to reveal the subspace structure of both source and target data through the locality aware reconstruction. This reconstruction keeps guiding the knowledge transfer in a latent shared subspace, and the whole learning process can be described as iterative updating reconstruction coefficients and subspace projections. In addition, researchers use a sparse error term to compensate for the outliers and therefore avoid negative transfer [16, 82, 15, 88]. On face, object, video benchmark databases, this line of methods achieve appealing results. Therefore, low-rank constraint based reconstruction has been considered as a very promising data alignment tool.

In this chapter, we propose a novel method targeting at the problem discussed above, called Missing Modality Transfer Learning via latent low-rank constraint (Fig. 3.2)[1] The main idea is to learn appropriate subspaces through a latent low-rank factor where data alignment is achieved both across modalities within one database, and across two databases. Therefore, the proposed transfer learning in two directions is promising to tackle the *Missing Modality Problem*.

We summarize our main contributions as follows:

- A novel transfer learning framework by extending conventional transfer learning into two directions is proposed to handle the *Missing Modality Problem*. By borrowing an auxiliary database with the same complete modalities, our M^{2}TL can learn appropriate low-dimensional subspaces from cross-modality direction and cross-database one.

- The latent low-rank model is incorporated into the transfer learning framework to uncover the missing modality with the existing modalities from two directions. The class structure information of the missing modality is uncovered from cross-modality direction, while the

[1] $X_{\mathrm{S\cdot B}}/X_{\mathrm{T\cdot B}}$ denote the source/target modalities in the object database B, where $X_{\mathrm{T\cdot B}}$ is also the missing modality. In addition, $X_{\mathrm{S\cdot A}}/X_{\mathrm{T\cdot A}}$ denote the source/target modalities from the auxiliary database A. Note in the illustration, same shape means same dataset and same color means same modality. The whole procedure is: introduce the auxiliary database A with modalities $X_{\mathrm{S\cdot A}}$ and $X_{\mathrm{T\cdot A}}$, and then transfer knowledge in two directions: **cross-modality transfer** ($\mathrm{T(M)}$) and **cross-database transfer** ($\mathrm{T(D)}$). In the unified model, P is the shared subspace projection, Y_{S} is pre-learned low-dimensional feature on the source domain X_{S}. The source and target domains are coupled by low-rank constraint Z and latent factor L. In addition, two datasets in the source domain are further coupled by Maximum Mean Discrepancy regularizer $\Omega(P) = \mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P)$.

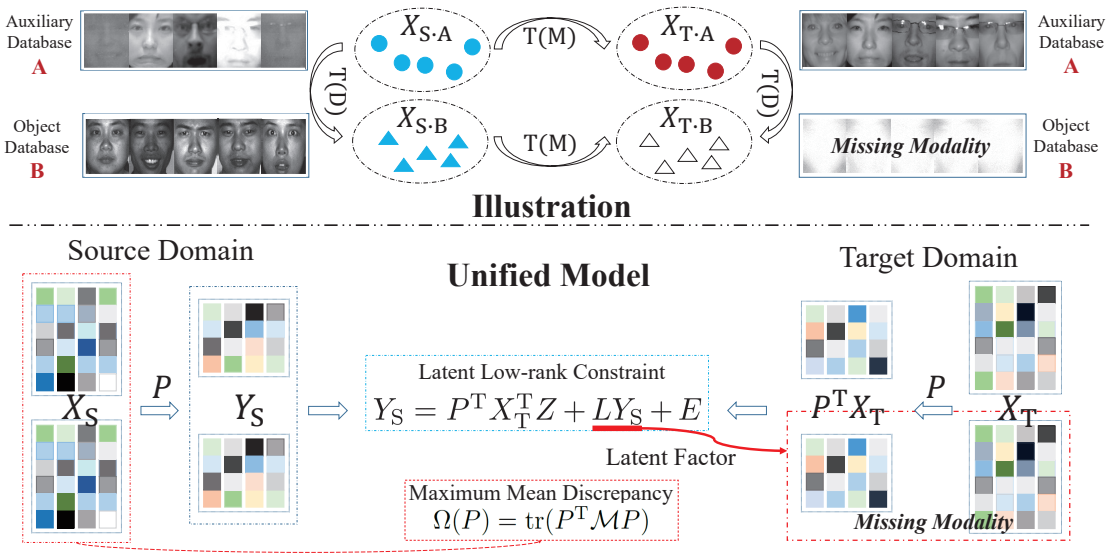


Figure 3.2: Illustration (above) and unified model (below) of our proposed M^{2}TL.

modality information is transferred from cross-database direction. In such a way, the uncovered latent information can assist in tackling the *Missing Modality Problem*.

- We develop an efficient solution to our proposed method with theoretical guarantees, by approximating the complex quadratic term with its first-order Taylor expansion.

3.2 Motivation

Given the auxiliary database A, and the object database B, each of which includes two modalities: $\{X_{\mathrm{S\cdot A}} \in \mathbb{R}^{d\times n_a}, X_{\mathrm{T\cdot A}} \in \mathbb{R}^{d\times n_a}\}$, and $\{X_{\mathrm{S\cdot B}} \in \mathbb{R}^{d\times n_b}, X_{\mathrm{T\cdot B}} \in \mathbb{R}^{d\times n_b}\}$, where d is the original feature dimensionality, n_a is the sample size of one modality in database A, whilst n_b is the sample size of one modality in database B. Let $n = n_a + n_b$, then we actually have four datasets in our *Missing Modality Problem* as Fig. 3.2 shows. Traditional transfer learning methods are interested in problems between different modalities within one database such as: $X_{\mathrm{S\cdot A}} \rightarrow X_{\mathrm{T\cdot A}}$, and $X_{\mathrm{S\cdot B}} \rightarrow X_{\mathrm{T\cdot B}}$, or between different databases within one modality such as: $X_{\mathrm{S\cdot A}} \rightarrow X_{\mathrm{S\cdot B}}$ and $X_{\mathrm{T\cdot A}} \rightarrow X_{\mathrm{T\cdot B}}$. Most importantly, those methods all require target data are available in the training stage. As stated in the *Missing Modality Problem*, this assumption is false. When $X_{\mathrm{T\cdot B}}$ is missing, most of transfer learning requiring accesses to both source and target data will fail.

Auxiliary database with exactly the same modalities as object database can help, but it introduces two new problems as well. First, how to transfer modalities knowledge from source to the target within database. Second, how to align the feature space of the auxiliary and object database. Both of them are critical to the *Missing Modality Problem*. Imagine we apply the transferred knowledge learned from ($X_{\mathrm{S\cdot A}} \rightarrow X_{\mathrm{T\cdot A}}$), to the testing data $X_{\mathrm{T\cdot B}}$ directly, the difference of marginal/conditional distributions between two databases A and B would ruin the performance, leading to a negative transfer. To that end, we have to consider transfer learning in two directions: one is knowledge transfer between modalities T(M); the other one is that between databases T(D), shown in Fig. 3.2. Learning in two directions is not independent any more, but rather closely related to each other to make sure right knowledge is passed on between both modalities and databases.

3.3 Transfer Learning via Latent Low-Rank Constraint

3.3.1 Conference Version Revisit

To address the *Missing Modality Problem*, we first project both source data X_{S} and target data X_{T} into some common subspace P that allows X_{S} and X_{T} to be aligned by low-rank constraint. Suppose projection P is known, both X_{S} and X_{T} are clean, and $X_{\mathrm{T\cdot B}}$ is observable, then the low-rank transfer subspace learning can be written as:

$$\min_{\widetilde{Z}} \|\widetilde{Z}\|_*, \ \text{s.t.} \ P^{\mathrm{T}} X_{\mathrm{S}} = P^{\mathrm{T}} X_{\mathrm{T}} \widetilde{Z}. \tag{3.1}$$

Assuming Eq. (3.1) has a unique solution, then we can derive that in subspace P, we have $P^{\mathrm{T}} X_{\mathrm{S}} \subseteq \mathrm{span}(P^{\mathrm{T}} X_{\mathrm{T}})$. Based on this result, we derive a new form for Eq. (3.1). Suppose $P^{\mathrm{T}}[X_{\mathrm{S}}, X_{\mathrm{T}}] = U\Sigma V^{\mathrm{T}}$ and $V = [V_{\mathrm{S}}; V_{\mathrm{T}}]$, where $P^{\mathrm{T}} X_{\mathrm{S}} = U\Sigma V_{\mathrm{S}}^{\mathrm{T}}$, $P^{\mathrm{T}} X_{\mathrm{T}} = U\Sigma V_{\mathrm{T}}^{\mathrm{T}}$. Then we can immediately deduct the constraint as $U\Sigma V_{\mathrm{S}}^{\mathrm{T}} = U\Sigma V_{\mathrm{T}}^{\mathrm{T}} \widetilde{Z}$. Therefore, Eq. (3.1) can be rewritten as:

$$\min_{\widetilde{Z}} \|\widetilde{Z}\|_*, \ \text{s.t.} \ V_{\mathrm{S}}^{\mathrm{T}} = V_{\mathrm{T}}^{\mathrm{T}} \widetilde{Z}. \tag{3.2}$$

According to **Theorem 3.1** [87], the optimal low-rank representation $\widetilde{Z}_*$ can be computed as:

$$\widetilde{Z}_* = V_{\mathrm{T}} V_{\mathrm{S}}^{\mathrm{T}} = [V_{\mathrm{T\cdot A}}; V_{\mathrm{T\cdot B}}] V_{\mathrm{S}}^{\mathrm{T}}, \tag{3.3}$$

where V_{T} has also been row partitioned into $V_{\mathrm{T\cdot A}}$ and $V_{\mathrm{T\cdot B}}$. The constrained part now can be rewritten as:

$$\begin{aligned} P^{\mathrm{T}}X_{\mathrm{S}} &= P^{\mathrm{T}}X_{\mathrm{T}}\widetilde{Z}_* = P^{\mathrm{T}}[X_{\mathrm{T\cdot A}}, X_{\mathrm{T\cdot B}}]\widetilde{Z}_* \\ &= P^{\mathrm{T}}[X_{\mathrm{T\cdot A}}, X_{\mathrm{T\cdot B}}][V_{\mathrm{T\cdot A}}; V_{\mathrm{T\cdot B}}]V_{\mathrm{S}}^{\mathrm{T}} \\ &= P^{\mathrm{T}}X_{\mathrm{T\cdot A}}(V_{\mathrm{T\cdot A}}V_{\mathrm{S}}^{\mathrm{T}}) + U\Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}}V_{\mathrm{T\cdot B}}V_{\mathrm{S}}^{\mathrm{T}} \\ &= P^{\mathrm{T}}X_{\mathrm{T\cdot A}}Z + (U\Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}}V_{\mathrm{T\cdot B}}\Sigma^{-1}U^{\mathrm{T}})P^{\mathrm{T}}X_{\mathrm{S}}, \end{aligned} \tag{3.4}$$

where $L = U\Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}}V_{\mathrm{T\cdot B}}\Sigma^{-1}U^{\mathrm{T}}$ should also be low-rank, as L can recover the structure of $P^{\mathrm{T}}X_{\mathrm{T\cdot B}}$.

From the above deduction, it is known that even $X_{\mathrm{T\cdot B}}$ is unobserved, we can recover it by imposing extra constraint:

$$\min_{Z,\,L} \|Z\|_* + \|L\|_*, \;\; \text{s.t.} \;\; P^{\mathrm{T}}X_{\mathrm{S}} = P^{\mathrm{T}}X_{\mathrm{T}}Z + LP^{\mathrm{T}}X_{\mathrm{S}}. \tag{3.5}$$

Therefore, the source data $P^{\mathrm{T}}X_{\mathrm{S}}$ is reconstructed from the column of $P^{\mathrm{T}}X_{\mathrm{T}}$ and the row of $P^{\mathrm{T}}X_{\mathrm{S}}$. When the target domain is missing some data, the row of $P^{\mathrm{T}}X_{\mathrm{S}}$ will make sense in reconstruction, uncovering its latent information.

3.3.2 Transfer Learning with Dictionary Constraint

For simplicity, we define the following three functions.

$$\boxed{\begin{array}{l} (1).\mathcal{L}(P, Z, L, E) = P^{\mathrm{T}}X_{\mathrm{S}} - P^{\mathrm{T}}X_{\mathrm{T}}Z - LP^{\mathrm{T}}X_{\mathrm{S}} - E \\ (2).\mathcal{D}(P, D, S) = \min_{D,S} \|P^{\mathrm{T}}X - DS\|_{\mathrm{F}}^2 + \gamma\|S\|_1 \\ (3).\mathcal{F}(Z, L, E) = \min_{Z,L,E} \|Z\|_* + \|L\|_* + \lambda\|E\|_{2,1} \end{array}}$$

We next integrate the subspace learning process into the above function. In general, subspace learning methods can be uniformed by the following:

$$\min_{P} \mathrm{tr}(P^{\mathrm{T}}\mathcal{W}P), \;\; \text{s.t.} \;\; P^{\mathrm{T}}\mathcal{U}P = \mathrm{I}, \tag{3.6}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operation. $\mathcal{W}$ and $\mathcal{U}$ are different defined according to the subspace learning methods.

Realistically, the data is often corrupted, so we add an error term E. Then the objective function of the general model can be rewritten as:

$$\begin{aligned} &\min_{P} \mathcal{F}(Z, L, E) + \psi\mathrm{tr}(P^{\mathrm{T}}\mathcal{W}P), \\ &\text{s.t.} \;\; \mathcal{L}(P, Z, L, E) = 0, \quad P^{\mathrm{T}}\mathcal{U}\mathrm{P} = \mathrm{I}, \end{aligned} \tag{3.7}$$

where we use $L_{2,1}$ norm on E to make it sample specific. $\psi > 0$ are parameters to balance the subspace part.

In addition, we introduce a common dictionary D on the projected data to further couple the knowledge from two domains. As a result, the dictionary and low-rank constraint on the projected data would work synchronously in optimizing the common subspace. This helps uncover the underlying structure of two domains, making our method more appropriate for the *Missing Modality Problem*. This process is illustrated in Figure 2, and the final objective function can be written formally as:

$$\begin{aligned} &\min_{P} \mathcal{F}(Z, L, E) + \psi \mathrm{tr}(P^{\mathrm{T}} \mathcal{W} P) + \varphi \mathcal{D}(P, D, S), \\ &\text{s.t. } \mathcal{L}(P, Z, L, E) = 0, \;\; P^{\mathrm{T}} \mathcal{U} \mathrm{P} = \mathrm{I}, \end{aligned} \tag{3.8}$$

where φ is the parameter that balances the influence of dictionary D. S represents sparse coefficients.

3.3.3 Low-Rank Transfer with Latent Factor

To recover $X_{\mathrm{T{\cdot}B}}$, we first assume it is observable, and then derive its formulation under our latent low-rank transfer learning framework. In the following section, we take cross-modality direction transfer T(M) as an example.

In conventional low-rank transfer subspace learning framework [16], source data can be reconstructed with a low-rank constraint in a common subspace $P \in \mathbb{R}^{d \times p}$, where p is the reduced feature dimensionality. In our problem, since we have two databases, each of which including both source and target data, we can formulate two parallel problems as: $X_{\mathrm{S{\cdot}A}} \rightarrow X_{\mathrm{T{\cdot}A}}$ and $X_{\mathrm{S{\cdot}B}} \rightarrow X_{\mathrm{T{\cdot}B}}$. $X_{\mathrm{S{\cdot}A}}$ and $X_{\mathrm{T{\cdot}A}}$ share a common subspace $P_{\mathrm{A}} \in \mathbb{R}^{d \times p}$ while $X_{\mathrm{S{\cdot}B}}$ and $X_{\mathrm{T{\cdot}B}}$ share another common subspace $P_{\mathrm{B}} \in \mathbb{R}^{d \times p}$. After projections, source/target modalities of object/auxiliary databases are lying in a more closed feature space.

Slightly different from our previous conference work [15] and LTSL [16], we introduce the pre-learned low-dimensional feature for the source data, meaning we only adapt the target modality in the transfer learning but keep the low-dimensional source feature fixed. By doing this, our framework is able to achieve much more stable solutions. Specifically, we pre-learn the low-dimensional feature $Y_{\mathrm{S{\cdot}A}} \in \mathbb{R}^{p \times n_a}$ and $Y_{\mathrm{S{\cdot}B}} \in \mathbb{R}^{p \times n_b}$ from the two source modalities from auxiliary and object databases by subspace learning methods, e.g.[64, 66, 65]. Then, we formulate our low-rank transfer learning with fixed pre-learned low-dimensional features for both auxiliary and object data as:

$$\min_{Z_{\mathrm{A}}} \mathrm{rank}(Z_{\mathrm{A}}), \;\; \text{s.t. } Y_{\mathrm{S{\cdot}A}} = P_{\mathrm{A}}^{\mathrm{T}} X_{\mathrm{T{\cdot}A}} Z_{\mathrm{A}}, \tag{3.9}$$

$$\min_{Z_{\mathrm{B}}} \mathrm{rank}(Z_{\mathrm{B}}), \;\; \text{s.t. } Y_{\mathrm{S{\cdot}B}} = P_{\mathrm{B}}^{\mathrm{T}} X_{\mathrm{T{\cdot}B}} Z_{\mathrm{B}}, \tag{3.10}$$

where $\mathrm{rank}(\cdot)$ represents the rank of a matrix. $Z_{\mathrm{A}} \in \mathbb{R}^{n_a \times n_a}$ and $Z_{\mathrm{B}} \in \mathbb{R}^{n_b \times n_b}$ are two low-rank coefficients matrixes. To couple the knowledge from two databases as well, we expect the projection P is common over two databases, namely, $P_{\mathrm{A}} = P_{\mathrm{B}} = P$. Therefore, we rewrite Eqs. (3.9)(3.10) into one objective function:

$$\min_{Z_{\mathrm{c}}} \mathrm{rank}(Z_{\mathrm{c}}), \;\; \text{s.t.} \;\; Y_{\mathrm{S}} = P^{\mathrm{T}} D_{\mathrm{T}} Z_{\mathrm{c}}, \tag{3.11}$$

where $Y_{\mathrm{S}} = [Y_{\mathrm{S\cdot A}}, Y_{\mathrm{S\cdot B}}]$, $D_{\mathrm{T}} = [X_{\mathrm{T\cdot A}}, X_{\mathrm{T\cdot B}}]$, and

$$Z_{\mathrm{c}} = \begin{bmatrix} Z_{\mathrm{A}} & 0 \\ 0 & Z_{\mathrm{B}} \end{bmatrix}.$$

Clearly, $\mathrm{rank}(Z_{\mathrm{c}}) = \mathrm{rank}(Z_{\mathrm{A}}) + \mathrm{rank}(Z_{\mathrm{B}})$; however, the $\mathrm{rank}(\cdot)$ minimization problem of either Z_{c} or $Z_{\mathrm{A/B}}$ is non-trivial to solve due to the non-convexity property. Recently, people use nuclear norm as a good surrogate for the rank minimization problem [49, 50], and achieve reasonable results. We then rewrite Eq. (3.11) as:

$$\min_{Z_{\mathrm{c}}} \|Z_{\mathrm{c}}\|_*, \;\; \text{s.t.} \;\; Y_{\mathrm{S}} = P^{\mathrm{T}} D_{\mathrm{T}} Z_{\mathrm{c}}, \tag{3.12}$$

where $\|\cdot\|_*$ is the nuclear norm of a matrix equal to the sum of singular values of the matrix.

Since Y_{S} can be spanned by $P^{\mathrm{T}} D_{\mathrm{T}}$, we could calculate $[Y_{\mathrm{S}}, P^{\mathrm{T}} D_{\mathrm{T}}] = U \Sigma V^{\mathrm{T}}$, where $V = [V_{\mathrm{S}}; V_{\mathrm{T}}]$ and $Y_{\mathrm{S}} = U \Sigma V_{\mathrm{S}}^{\mathrm{T}}$, $P^{\mathrm{T}} D_{\mathrm{T}} = U \Sigma V_{\mathrm{T}}^{\mathrm{T}}$. Then we can immediately deduct the constraint as $U \Sigma V_{\mathrm{S}}^{\mathrm{T}} = U \Sigma V_{\mathrm{T}}^{\mathrm{T}} Z_{\mathrm{c}}$. Therefore, Eq. (3.12) can be further rewritten as:

$$\min_{Z_{\mathrm{c}}} \|Z_{\mathrm{c}}\|_*, \;\; \text{s.t.} \;\; V_{\mathrm{S}}^{\mathrm{T}} = V_{\mathrm{T}}^{\mathrm{T}} Z_{\mathrm{c}}, \tag{3.13}$$

where optimal low-rank representation Z_{c}^* can be computed as: (**Theorem 3.1** in [87]):

$$Z_{\mathrm{c}}^* = V_{\mathrm{T}} V_{\mathrm{S}}^{\mathrm{T}} = \begin{bmatrix} V_{\mathrm{T\cdot A}} \\ V_{\mathrm{T\cdot B}} \end{bmatrix} V_{\mathrm{S}}^{\mathrm{T}}, \tag{3.14}$$

where V_{T} has also been vertically partitioned into $V_{\mathrm{T\cdot A}}$ and $V_{\mathrm{T\cdot B}}$. By inserting this into Eq. (3.12), we subsequently get the following deduction:

$$
\begin{aligned}
Y_{\mathrm{S}} &= P^{\mathrm{T}} D_{\mathrm{T}} Z_{\mathrm{c}}^{*} = [P^{\mathrm{T}} X_{\mathrm{T\cdot A}}, P^{\mathrm{T}} X_{\mathrm{T\cdot B}}] Z_{\mathrm{c}}^{*} \\
&= [P^{\mathrm{T}} X_{\mathrm{T\cdot A}}, P^{\mathrm{T}} X_{\mathrm{T\cdot B}}] \begin{bmatrix} V_{\mathrm{T\cdot A}} \\ V_{\mathrm{T\cdot B}} \end{bmatrix} V_{\mathrm{S}}^{\mathrm{T}} \\
&= P^{\mathrm{T}} X_{\mathrm{T\cdot A}} (V_{\mathrm{T\cdot A}} V_{\mathrm{S}}^{\mathrm{T}}) + P^{\mathrm{T}} X_{\mathrm{T\cdot B}} V_{\mathrm{T\cdot B}} V_{\mathrm{S}}^{\mathrm{T}} \\
&= P^{\mathrm{T}} X_{\mathrm{T\cdot A}} (V_{\mathrm{T\cdot A}} V_{\mathrm{S}}^{\mathrm{T}}) + U \Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}} V_{\mathrm{T\cdot B}} V_{\mathrm{S}}^{\mathrm{T}} \\
&= P^{\mathrm{T}} X_{\mathrm{T}} Z + (U \Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}} V_{\mathrm{T\cdot B}} \Sigma^{-1} U^{\mathrm{T}}) Y_{\mathrm{S}} \\
&= P^{\mathrm{T}} X_{\mathrm{T}} Z + L Y_{\mathrm{S}},
\end{aligned}
\tag{3.15}
$$

where $Z = V_{\mathrm{T\cdot A}} V_{\mathrm{S}}^{\mathrm{T}}$, $L = U \Sigma V_{\mathrm{T\cdot B}}^{\mathrm{T}} V_{\mathrm{T\cdot B}} \Sigma^{-1} U^{\mathrm{T}}$, and $X_{\mathrm{T}} = X_{\mathrm{T\cdot A}}$.

It should be indicated that based on Eq. (3.14), both $Z \in \mathbb{R}^{n_a \times n}$ and $L \in \mathbb{R}^{p \times p}$ are inclined to be low-rank, which casts a light on recovering the missing data $X_{\mathrm{T\cdot B}}$. To be concrete, since $X_{\mathrm{T\cdot B}}$ is a factor in L's formulation, optimizing over L will consequently optimize over $X_{\mathrm{T\cdot B}}$, which in turn recovers the missing data. In brief, we are able to recover the latent factor based on the following new formulation optimized over both Z and L:

$$
\begin{aligned}
&\min_{Z,\, L} \|Z\|_{*} + \|L\|_{*}, \\
&\text{s.t. } Y_{\mathrm{S}} = P^{\mathrm{T}} X_{\mathrm{T}} Z + L Y_{\mathrm{S}},
\end{aligned}
\tag{3.16}
$$

Next we give some insights about the latent low-rank transfer learning proposed above: (1) Latent low-rank constraint $Y_{\mathrm{S}} = P^{\mathrm{T}} X_{\mathrm{T}} Z + L Y_{\mathrm{S}}$ essentially unifies previous low-rank transfer learning methods [82, 16]. We reformulate this constraint into $(\mathrm{I_p} - L) Y_{\mathrm{S}} = P^{\mathrm{T}} X_{\mathrm{T}} Z$, $\mathrm{I_p} \in \mathbb{R}^{p \times p}$, and easily find that the transformed low-dimensional feature of source data $(\mathrm{I_p} - L) Y_{\mathrm{S}}$ is reconstructed by the projected target data $P^{\mathrm{T}} X_{\mathrm{T}}$, which integrates both subspace learning from [16], and feature rotation from [82]. (2) It indicates that the introduction of pre-learned low-dimensional feature for the source data is reasonable, since the latent factor will adjust the source feature anyhow in our formulation. (3) Source data Y_{S} is reconstructed from both column space of $P^{\mathrm{T}} X_{\mathrm{T}}$ and the row space of Y_{S}. This is especially useful when target data of object database are missing from X_{T}, giving rise to incomplete column space of $P^{\mathrm{T}} X_{\mathrm{T}}$.

3.3.3.1 Learning Projections

Although latent low-rank constraint has many advantages over conventional methods, it does not explicitly model the relation between two datasets in the source domain, e.g., $X_{\mathrm{S\cdot A}}$, $X_{\mathrm{S\cdot B}}$.

Therefore we introduce a regularizer to enforce their correlations. A straightforward way is to 'push' the means of two datasets closer, namely, minimizing the following problem:

$$\begin{aligned}\Omega(P) &= \|\frac{1}{n_a}\sum_{i=1}^{n_a} P^{\mathrm{T}}x_i - \frac{1}{n_b}\sum_{j=n_a+1}^{n} P^{\mathrm{T}}x_j\|_{\mathrm{F}}^2 \\ &= \|P^{\mathrm{T}}\mu_{\mathrm{A}} - P^{\mathrm{T}}\mu_{\mathrm{B}}\|_{\mathrm{F}}^2 \\ &= \mathrm{tr}(P^{\mathrm{T}}(\mu_{\mathrm{A}} - \mu_{\mathrm{B}})(\mu_{\mathrm{A}} - \mu_{\mathrm{B}})^{\mathrm{T}}P) \\ &= \mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P)\end{aligned} \tag{3.17}$$

where $\|\cdot\|_{\mathrm{F}}^2$ is matrix Frobenius norm, x_i and x_j are source data from different databases, $\mu_{\mathrm{A}} = \frac{1}{n_a}\sum_{i=1}^{n_a} x_i$ and $\mu_{\mathrm{B}} = \frac{1}{n_b}\sum_{j=n_a+1}^{n} x_j$, $\mathrm{tr}(\cdot)$ is the trace of a matrix. The above regularizer has been studied in [89, 90, 81] and shows promising results under transfer learning scenario. Here, we adopt it for the alignment of two datasets from the source domain, rather than across different domains. Note that we introduce pre-learned low-dimensional feature of source data into the latent low-rank constraint, and the pre-learned features of two source subsets would have different distributions. The projected source data $P^{\mathrm{T}}X_{\mathrm{S\cdot A}}$ and $P^{\mathrm{T}}X_{\mathrm{S\cdot B}}$ are the new representations, which intend to be close to the pre-learned one $Y_{\mathrm{S\cdot A}}$ and $Y_{\mathrm{S\cdot B}}$. Our regularizer would enforce the mean of two projected source subsets close together, that is, the projected source data are the bridge between the pre-learned low-dimensional features. The learned projection and the new low-dimensional representations are essentially compromise between the two constraints. Solutions meet only one condition may conflict with another, but we jointly optimize under two constraints in one problem.

We further relax the original problem by adding a term $E \in \mathbb{R}^{p\times n}$ to the latent low-rank constraint. This benefits our model in two folds. First, it transforms the original hard constraint to a soft one, which avoids the potential over-fitting problem. Second, in practice, term E is able to compensate for the data noise if we minimize its l_1-norm at the same time [49, 87].

Moreover, to make the learned P more effective, we introduce the group structure sparsity to select the most important features as the same time with subspace learning. Therefore, the objective function of the latent low-rank transfer learning can be rewritten as:

$$\begin{aligned}&\min_{P,Z,L,E} \|Z\|_* + \|L\|_* + \lambda\|E\|_1 + \alpha\|P\|_{2,1} + \beta\Omega(P), \\ &\text{s.t. } Y_{\mathrm{S}} = P^{\mathrm{T}}X_{\mathrm{T}}Z + LY_{\mathrm{S}} + E, P^{\mathrm{T}}P = \mathrm{I}_{\mathrm{p}},\end{aligned} \tag{3.18}$$

where λ, α and β are three balance parameters. Note the orthogonal constraint $P^{\mathrm{T}}P = \mathrm{I}_{\mathrm{p}}$ ($\mathrm{I}_{\mathrm{p}} \in \mathbb{R}^{p\times p}$) is imposed to avoid arbitrary small trivial solutions of subspace P.

Discussion: In transfer learning, both marginal and conditional distributions are critical for performance. Similar marginal distributions ($\Pr(X_{\mathrm{S}}) \approx \Pr(X_{\mathrm{T}})$) indicates both source and target data lie in the same feature space, while similar conditional distributions ($\Pr(Y|X_{\mathrm{S}}) \approx \Pr(Y|X_{\mathrm{T}})$) guarantee that discriminative power can be passed on from source to target domains. In *Missing Modality Problem*, source and target domains are drawn from different distributions, meaning $\Pr(X_{\mathrm{S}}) \neq \Pr(X_{\mathrm{T}})$. Fortunately, the learned common subspace P is able to mitigate the divergence in the sense of $\Pr(P^{\mathrm{T}}X_{\mathrm{S}}) \approx \Pr(P^{\mathrm{T}}X_{\mathrm{T}})$.

On the other hand, adapting conditional distributions of one or two of them is not an easy task. In reality, it is often the case that the number of classes are different between source and target data. Under condition, either some source knowledge may become redundant, or target data are lack of sufficient knowledge. In fact, even different number of classes will help [16], if we consider the latent low-rank transfer learning as many-to-many mapping, meaning each class of source data are essentially built by data from a few classes in the target domain, and a specific class of data in the target domain may be correlated with different classes in the source domain. This is essentially a 'coarse' version of the original conditional distributions, namely, $\Pr(\hat{Y}|X_{\mathrm{S}}) \approx \Pr(\tilde{Y}|X_{\mathrm{T}})$, where $\hat{Y}$ and $\tilde{Y}$ are new label sets of source and target data, by merging labels from Y.

3.3.3.2 Solving the Optimization Problem

Problem (3.18) could be solved by off-the-shelf algorithms, e.g., Augmented Lagrange Methods (ALM) [49, 87]. However, extra relax variables in ALM lead to complex matrix operations, e.g., inverse, multiplications, in each iteration. This is essentially caused by the quadratic term in the augmented Lagrangian function, which includes linear mappings of the target variables. To reduce the computation cost of this part, we propose to use the first order Taylor expansion like approximation to replace the original quadratic term, leading to a simpler solution to the original problem. To make it clear, we first write down the augmented Lagrangian function of problem (3.18):

$$\begin{aligned} &\|Z\|_* + \|L\|_* + \lambda\|E\|_1 + \alpha\|P\|_{2,1} + \beta\mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P) \\ &+\langle Y_1, Y_{\mathrm{S}} - P^{\mathrm{T}}X_{\mathrm{T}}Z - LY_{\mathrm{S}} - E\rangle \\ &+\frac{\mu}{2}(\|Y_{\mathrm{S}} - P^{\mathrm{T}}X_{\mathrm{T}}Z - LY_{\mathrm{S}} - E\|_{\mathrm{F}}^2), \end{aligned} \tag{3.19}$$

where Y_1 is the lagrange multiplier and $\mu > 0$ is a penalty parameter. $\langle , \rangle$ is the inner product of matrixes and $\langle A, B\rangle = \mathrm{tr}(A^{\mathrm{T}}B)$. We then merge the last two terms into quadratic terms, and

formulate it as:

$$\begin{aligned}&\|Z\|_* + \|L\|_* + \lambda\|E\|_1 + \alpha\|P\|_{2,1} + \beta\mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P)\\&+h(Z, L, E, P, Y_1, \mu) - \frac{1}{\mu}\|Y_1\|_{\mathrm{F}}^2,\end{aligned} \tag{3.20}$$

where $h(Z, L, E, P, Y_1, \mu) = \frac{\mu}{2}(\|Y_{\mathrm{S}} - P^{\mathrm{T}}X_{\mathrm{T}}Z - LY_{\mathrm{S}} - E + Y_1/\mu\|_{\mathrm{F}}^2)$. Like the conventional ALM, the new formulation is not jointly solvable over Z, L, E and P, but solvable over each of them, by fixing rest of them. Therefore, we solve each subproblem at a time, and approximate the quadratic term h with first order expansion at the current point, assuming others are constant. At iteration $t+1$ ($t \geq 0$), we have:

Updating Z:

$$\begin{aligned}Z^{(t+1)} &= \arg\min_Z \|Z\|_* + h(Z, L^{(t)}, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu)\\&= \arg\min_Z \|Z\|_* + \frac{\eta_z\mu}{2}\|Z - Z^{(t)}\|_{\mathrm{F}}^2\\&+\langle\nabla_Z h(Z^{(t)}, L^{(t)}, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu), Z - Z^{(t)}\rangle\\&= \arg\min_Z \frac{1}{\eta_z\mu}\|Z\|_* + \frac{1}{2}\|Z - Z^{(t)} + \nabla_Z h\|_{\mathrm{F}}^2\end{aligned} \tag{3.21}$$

where $\nabla_Z h = \nabla_Z h(Z^{(t)}, L^{(t)}, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu) = X_{\mathrm{T}}^{\mathrm{T}}P^{(t)}(Y_{\mathrm{S}} - {P^{(t)}}^{\mathrm{T}}X_{\mathrm{T}}Z^{(t)} - L^{(t)}Y_{\mathrm{S}} - E^{(t)} + Y_1^{(t)}/\mu)$ and $\eta_z = \|{P^{(t)}}^{\mathrm{T}}X_{\mathrm{T}}\|_2^2$. Problem (3.21) can be effectively solved by the singular value thresholding (SVT) operator [60]. Define $U_Z\Sigma_Z V_Z$ as the SVD of matrix $(Z^{(t)} - \nabla_Z h)$, where $\Sigma_Z = \mathrm{diag}(\{\sigma_i\}_{1\leq i\leq r})$, σ_i is the singular value and r is the rank. Then, the optimal $Z^{t+1} = U_Z\Omega_{(\frac{1}{\mu})}(\Sigma_Z)V_Z$, where $\Omega_{(\frac{1}{\mu})} = \mathrm{diag}(\{\sigma_i - \frac{1}{\mu}\}_+)$, and q_+ means the positive part of q [60].

Updating L:

$$\begin{aligned}L^{(t+1)} &= \arg\min_L \|L\|_* + h(Z^{(t+1)}, L, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu)\\&= \arg\min_L \|L\|_* + \frac{\eta_l\mu}{2}\|L - L^{(t)}\|_{\mathrm{F}}^2\\&+\langle\nabla_L h(Z^{(t)}, L^{(t)}, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu), L - L^{(t)}\rangle\\&= \arg\min_L \frac{1}{\eta_l\mu}\|L\|_* + \frac{1}{2}\|L - L^{(t)} + \nabla_L h\|_{\mathrm{F}}^2\end{aligned} \tag{3.22}$$

where $\nabla_L h = \nabla_L h(Z^{(t+1)}, L^{(t)}, E^{(t)}, P^{(t)}, Y_1^{(t)}, \mu) = (Y_{\mathrm{S}} - {P^{(t)}}^{\mathrm{T}}X_{\mathrm{T}}Z^{(t+1)} - L^{(t)}Y_{\mathrm{S}} - E^{(t)} + Y_1^{(t)}/\mu)Y_{\mathrm{S}}^{\mathrm{T}}$ and $\eta_l = \|Y_{\mathrm{S}}\|_2^2$. Problem (3.22) can be solved via the singular value thresholding (SVT) operator [60] in the same way as (3.21).

Updating E:

$$\begin{aligned} E^{(t+1)} = \arg\min_{E} \frac{\lambda}{\mu}\|E\|_1 + \frac{1}{2}\|E - (Y_{\mathrm{S}} \\ -P^{(t)^{\mathrm{T}}} X_{\mathrm{T}} Z^{(t+1)} - L^{(t+1)} Y_S + Y_1^{(t)}/\mu)\|_{\mathrm{F}}^2 \end{aligned} \tag{3.23}$$

which is solved by the shrinkage operator [91].

Updating P:

$$\begin{aligned} P^{(t+1)} = \arg\min_{P} \alpha\|P\|_{2,1} + \beta \mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P) + \frac{\mu}{2}(\|Y_{\mathrm{S}} - \\ P^{\mathrm{T}} X_{\mathrm{T}} Z^{(t+1)} - L^{(t+1)} Y_{\mathrm{S}} - E^{(t+1)} + Y_1^{(t)}/\mu\|_{\mathrm{F}}^2 \end{aligned}$$

which is transformed into the equivalent problem [92] as:

$$\begin{aligned} P^{(t+1)} = \arg\min_{P} \alpha \mathrm{tr}(P^{\mathrm{T}} G^{(t)} P) + \beta \mathrm{tr}(P^{\mathrm{T}}\mathcal{M}P) + \frac{\mu}{2}(\|Y_{\mathrm{S}} \\ -P^{\mathrm{T}} X_{\mathrm{T}} Z^{(t+1)} - L^{(t+1)} Y_{\mathrm{S}} - E^{(t+1)} + Y_1^{(t)}/\mu\|_{\mathrm{F}}^2 \end{aligned}$$

where $G^{(t)}$ is a diagonal matrix with the j-th diagonal element equal to

$$g_{jj}^{(t)} = \begin{cases} 0, & \text{if } p_j^{(t)} = \mathbf{0}, \\ \frac{1}{2\|p_j^{(t)}\|_2}, & \text{otherwise.} \end{cases}$$

and $p_j^{(t)}$ is the j-th row of $P^{(t)}$. Therefore, we can achieve

$$\begin{aligned} P^{(t+1)} \quad &= (2\alpha G^{(t)} + 2\beta\mathcal{M} + \mu X_{\mathrm{T}} Z^{(t+1)} (X_{\mathrm{T}} Z^{(t+1)})^{\mathrm{T}})^{-1} \\ &\mu X_{\mathrm{T}} Z^{(t+1)} (Y_{\mathrm{S}} - L^{(t+1)} Y_{\mathrm{S}} - E^{(t+1)} + Y_1^{(t)}/\mu)^{\mathrm{T}} \end{aligned} \tag{3.24}$$

The whole procedure of our solutions is outlined in **Algorithm 1**. And the parameters μ, ρ, ϵ, $\max_\mu$ and maxIter are set empirically, while other balanced parameters α, β, λ are tuned in the experiment.

3.3.3.3 Complexity and Convergency

For simplicity, assume X_{S} and X_{T} are both $d \times n$ matrixes, and P is a $d \times p$ matrix, where d is the original feature dimensionality, n is the size of source and target, and p is the reduced dimensionality. Then time-consuming components of **Algorithm 1**: 1). Trace norm computation in Step 1 and 2; 2). Matrix multiplication and inverse in Step 4.

Algorithm 1 Solving Problem (3.19)

Input: $X_{\mathrm{S}}, X_{\mathrm{T}}, \lambda, \alpha, \beta, \mathcal{M}, Y_{\mathrm{S}}$

Initialize: $Z^{(0)} = 0, J^{(0)} = 0, E^{(0)} = 0, Y_1^{(0)} = 0, \epsilon = 10^{-6}$,
$\mu = 10^{-6}, \rho = 1.2, \max_\mu = 10^6, \mathrm{maxIter} = 50, t = 0.$

while not converged **or** $t \leq$ maxIter **do**

1. Fix the others and update $Z^{(t+1)}$ according to (3.21);
2. Fix the others and update $L^{(t+1)}$ according to (3.22);
3. Fix the others and update $E^{(t+1)}$ according to (3.23);
4. Fix the others and update $P^{(t+1)}$ according to (3.24), then $P^{(t+1)} \leftarrow \mathbf{orth}(P^{(t+1)})$
5. Update the multipliers $Y_1^{(t+1)}$
   $Y_1^{(t+1)} = Y_1^{(t)} + \mu(Y_{\mathrm{S}} - {P^{(t+1)}}^{\mathrm{T}} X_{\mathrm{T}} Z^{(t+1)} - L^{(t+1)} Y_{\mathrm{S}} - E^{(t+1)})$;
6. Update the parameter μ by $\mu = \min(\rho\mu, \max_\mu)$;
7. Check the convergence conditions
   $\|Y_{\mathrm{S}} - {P^{(t+1)}}^{\mathrm{T}} X_{\mathrm{T}} Z^{(t+1)} - L^{(t+1)} Y_{\mathrm{S}} - E^{(t+1)}\|_\infty < \epsilon$.
8. $t = t + 1$.

end while

output: Z, L, E, P

Here, we discuss the computation complexity in detail. The SVD computation in Step 1 takes $O(n^3)$, and that in Step 2 takes $O(p^3)$. In fact, Step 2 is very fast as the dimension of the projected space is very low, while Step 1 would cost a lot when the size of dataset is very large, but this can be improved to $O(rn^2)$ by accelerations of SVD, where r is the rank of the low-rank matrix. The general multiplication each takes $O(d^3)$ and the inverse also costs $O(d^3)$ for $d \times d$ matrixes. Due to there are l multiplications, Step 4 costs nearly $(l+1)O(d^3)$. Next, we theoretically demonstrate that the proposed efficient algorithm will converge to a local minima and the convergence speed is affected by the perturbation caused by projections on the manifold during the alteration projection process. We first introduce the notation which is used in the convergence proof.

Notation: $\mathcal{P}_{\mathcal{Z}}$ is the operator to calculate $\{L, E\}$ using Z, $\mathcal{P}_{\mathcal{L}}$ is the operator to calculate $\{Z, E\}$ using L and $\mathcal{P}_{\mathcal{E}}$ is the operator to calculate $\{Z, L\}$ using E. $\widetilde{Z} = P_1^\dagger(LY_{\mathrm{S}}+E)$, $\widetilde{L} = (P^{\mathrm{T}}X_{\mathrm{T}}Z+E)P_2^\dagger$ and $\widetilde{E} = P^{\mathrm{T}}X_{\mathrm{T}}Z + LY_{\mathrm{S}}$. $P_1^\dagger$ and $P_2^\dagger$ are the pseudo-inverses of $P^{\mathrm{T}}X_{\mathrm{T}}$ and Y_{S}.

Theorem 1. $\|Y_{\mathrm{S}} - P^{\mathrm{T}}X_{\mathrm{T}}Z - LY_{\mathrm{S}} - E\|_{\mathrm{F}}^2$ converges to a local minimum when P is fixed. And the asymptotical and convergence speed of $\{Z, L, E\}$ will be accelerated by shrinking: 1) $\|\Delta_Z\|_{\mathrm{F}}/\|Z + \Delta_Z\|_{\mathrm{F}}$ for Z, where $\Delta_Z = \widetilde{Z} + \mathcal{P}_{\mathcal{Z}}(\widetilde{Z})$; 2) $\|\Delta_L\|_{\mathrm{F}}/\|L + \Delta_L\|_{\mathrm{F}}$ for L, where $\Delta_L = \widetilde{L} + \mathcal{P}_{\mathcal{L}}(\widetilde{L})$; 3) $\|\Delta_E\|_{\mathrm{F}}/\|E + \Delta_E\|_{\mathrm{F}}$ for E, where $\Delta_E = \widetilde{E} + \mathcal{P}_{\mathcal{E}}(\widetilde{E})$.

Proof: First, we prove that the constraint $\|\mathcal{C}(P,Z,L,E)\|_{\mathrm{F}}^2$ converges to a local minimum when P is fixed. We define the reconstruct error O_t^1, O_t^2, O_t^3 respectively for three variables Z, L, E in t^{th} iteration.

For Z,

$$\begin{cases} O_t^1 = \|\mathcal{C}(P, Z_{t-1}, L_{t-1}, E_{t-1})\|_{\mathrm{F}}^2, \\ O_t^2 = \|\mathcal{C}(P, Z_t, L_{t-1}, E_{t-1})\|_{\mathrm{F}}^2. \end{cases} \tag{3.25}$$

The global optimality of Z_t produces $O_t^1 \geq O_t^2$.

For L,

$$\begin{cases} O_t^2 = \|\mathcal{C}(P, Z_{t-1}, L_{t-1}, E_{t-1})\|_{\mathrm{F}}^2, \\ O_t^3 = \|\mathcal{C}(P, Z_t, L_t, E_{t-1})\|_{\mathrm{F}}^2. \end{cases} \tag{3.26}$$

The global optimality of L_t produces $O_t^2 \geq O_t^3$.

For E,

$$\begin{cases} O_t^3 = \|\mathcal{C}(P, Z_t, L_t, E_{t-1})\|_{\mathrm{F}}^2, \\ O_{t+1}^1 = \|\mathcal{C}(P, Z_t, L_t, E_t)\|_{\mathrm{F}}^2. \end{cases} \tag{3.27}$$

The global optimality of E_t produces $O_t^3 \geq O_{t+1}^1$. Therefore, the low-rank constraint $\|\mathcal{C}(P,Z,L,E)\|_{\mathrm{F}}^2$ keep decreasing in our algorithm:

$$O_1^1 \geq O_1^2 \geq O_1^3 \geq O_2^1 \geq \cdots \geq O_t^3 \geq O_{t+1}^1 \cdots \tag{3.28}$$

This completes the proof that the low-rank constraint converges to a local minimum when solving each variable using our proposed solution.

Next, we prove the asymptotical and convergence speed of $\{Z, L, E\}$, which can be demonstrated via *alternating projections on manifolds* [93]. Let's first consider Z. Take the (t+1)th iteration for example. We have

$$\begin{aligned} Z_{t+1} &= \mathcal{P}_{\mathcal{M}}(P_1^{\dagger} Y_{\mathrm{S}} - \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger} Y_{\mathrm{S}} - Z_t)) \\ &= \mathcal{P}_{\mathcal{M}} \mathcal{P}_{\mathcal{N}}(Z_t), \end{aligned} \tag{3.29}$$

where $\mathcal{P}_{\mathcal{Z}}$ is the operator to calculate $\{L_t, E_t\}$ using Z_t. $\mathcal{M}$ and $\mathcal{N}$ are two C^k-manifolds around a point $\overline{Z} \in \mathcal{M} \cap \mathcal{N}$:

$$\begin{cases} \mathcal{M} = \{\widehat{Z} \in \mathbb{R}^{n \times n}\}, \\ \mathcal{N} \ = \{P_1^{\dagger} Y_{\mathrm{S}} - \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger} Y_{\mathrm{S}} - \widehat{Z}), \widehat{Z} \in \mathbb{R}^{n \times n}\}. \end{cases} \tag{3.30}$$

The angle of two manifolds $\mathcal{M}$ and $\mathcal{N}$ at point Z is defined as:

$$\begin{aligned} c(\mathcal{M},\mathcal{N},Z) = \max\{\langle x,z\rangle : x \in \mathbb{S}\cap T_{\mathcal{M}}(Z)\cap N_{\mathcal{N}}(Z), \\ z \in \mathbb{S}\cap T_{\mathcal{N}}(Z)\cap N_{\mathcal{M}}(Z)\}, \end{aligned} \tag{3.31}$$

where $T_{\mathcal{M}}(Z)$ and $T_{\mathcal{N}}(Z)$ are the tangent space of manifolds $\mathcal{M}$ and $\mathcal{N}$ on point Z, while $N_{\mathcal{M}}(Z)$ and $N_{\mathcal{N}}(Z)$ are the normal space. $\mathbb{S}$ is the unit sphere.

According to **Theorems** (2-4) [94], $c(\mathcal{M},\mathcal{N},\overline{Z})$, which controls the asymptotic and convergence speed, is influenced by Z, L, E. Then, we give the detail how the three variables influence the asymptotic and convergence speed.

The normal spaces of manifolds $\mathcal{M}$ and $\mathcal{N}$ on point $\overline{Z}$ are respectively defined as

$$\begin{cases} N_{\mathcal{M}}(\overline{Z}) = \{z : u_i^{\mathrm{T}} z v_i = 0, \overline{Z} = UDV^{\mathrm{T}}\}, \\ N_{\mathcal{N}}(\overline{Z}) = \{P_1^{\dagger} Y_{\mathrm{S}} - \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger} Y_{\mathrm{S}} - \overline{Z})\}, \end{cases} \tag{3.32}$$

Assume $P_1^{\dagger} Y_{\mathrm{S}} = \overline{Z} + P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E})$ in the converged state. Then, from the normal space of manifolds $\mathcal{N}$, we get

$$\widehat{Z} = P_1^{\dagger} Y_{\mathrm{S}} - \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E})). \tag{3.33}$$

Therefore, we achieve

$$\begin{aligned} \widehat{Z} &= \overline{Z} + P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E}) - \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E})) \\ &= \overline{Z} + \Delta_Z, \end{aligned} \tag{3.34}$$

where $\Delta_Z = P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E}) + \mathcal{P}_{\mathcal{Z}}(P_1^{\dagger}(\overline{L} Y_{\mathrm{S}} + \overline{E})) = \widetilde{Z} + \mathcal{P}_{\mathcal{Z}}(\widetilde{Z})$, which can be treated as the control factor of L and E in updating Z. Thus, the normal space of manifold $\mathcal{N}$ can be rewritten as $N_{\mathcal{N}}(\overline{Z}) = \{\overline{Z} + \Delta_Z\}$.

Due to the tangent space and normal space are complementary, so we can derive that $N_{\mathcal{N}}(\overline{Z}) \subseteq T_{\mathcal{M}}(\overline{Z})$ and $N_{\mathcal{M}}(\overline{Z}) \subseteq T_{\mathcal{N}}(\overline{Z})$. Then Eq. (3.31) can be simplified as

$$\begin{aligned} c(\mathcal{M},\mathcal{N},\overline{Z}) = \max\{\langle x,z\rangle : x \in \mathbb{S}\cap N_{\mathcal{N}}(\overline{Z}), \\ z \in \mathbb{S}\cap N_{\mathcal{M}}(\overline{Z})\}. \end{aligned} \tag{3.35}$$

Therefore, we achieve

$$\begin{aligned} \langle x,z\rangle &= \mathrm{tr}(VDU^{\mathrm{T}} z + \Delta_Z^{\mathrm{T}} z) \\ &= \mathrm{tr}(VDU^{\mathrm{T}} z) + \mathrm{tr}(\Delta_Z^{\mathrm{T}} z) \\ &= \mathrm{tr}(\Delta_Z^{\mathrm{T}} z) \end{aligned} \tag{3.36}$$

where $\mathrm{tr}(VDU^{\mathrm{T}}z) = \mathrm{tr}(DU^{\mathrm{T}}zV) = \sum_i Du_i^{\mathrm{T}}zv_i = 0$, as $u_i^{\mathrm{T}}zv_i = 0$. Then,

$$\begin{aligned} c(\mathcal{M}, \mathcal{N}, \overline{Z}) &= \max\{\langle x, z\rangle\} \leq \max\{\langle D_{\Delta_Z}, D_z\rangle\} \\ &\leq \|D_{\Delta_Z}\|_{\mathrm{F}}\|D_z\|_{\mathrm{F}} \leq \|D_{\Delta_Z}\|_{\mathrm{F}} \end{aligned} \tag{3.37}$$

where the diagonal entries of D_{Δ_Z} and D_z are the eigenvalues of Δ_Z and z. Therefore, the asymptotic and convergence speeds of Z will be accelerated by shrinking $\|\Delta_Z\|_{\mathrm{F}}$, and vice versa. In general, $(Z + \Delta_Z)$ is not normalized onto the sphere $\mathbb{S}$, therefore, $\|\Delta_Z\|_{\mathrm{F}}$ should be substituted by $\|\Delta_Z\|_{\mathrm{F}}/\|Z + \Delta_Z\|_{\mathrm{F}}$.

For variables L and E, we can also use the similar way to prove. We can achieve: for L, $\|\Delta_L\|_{\mathrm{F}}/\|L+\Delta_L\|_{\mathrm{F}}$, where $\Delta_L = \widetilde{L}+\mathcal{P}_{\mathcal{L}}(\widetilde{L})$; for E, $\|\Delta_E\|_{\mathrm{F}}/\|E+\Delta_E\|_{\mathrm{F}}$, where $\Delta_E = \widetilde{E}+\mathcal{P}_{\mathcal{E}}(\widetilde{E})$. Therefore, we complete the whole proof.

3.3.3.4 Transfer in Two Directions

In this section, we extend the proposed latent low-rank transfer learning model to two directions (Fig. 3.2). Recall that our model in Section 3.2 is designed for mitigating the distributions of source and target data in one direction, meanwhile minimizing the divergence of two datasets in the source domain with a regularizer. However, the *Missing Modality Problem* involves two databases, each with two modalities. In fact, the auxiliary database A promises the similar modality configuration compared to the objective one B, but is not captured under exactly the same situation. Therefore, it is not enough to only consider transferring knowledge between two modalities in the auxiliary database, as the general transfer learning algorithms do. The proposed two directional transfer learning allows the knowledge transferred between databases as well, which in turn mitigates the divergence between two databases. Meanwhile, the latent factor still works in the new direction, and the regularizer couples the knowledge from the two datasets of the source domain.

From Fig. 3.2, we can observe that missing modality $X_{\mathrm{T\cdot B}}$ is more related with $X_{\mathrm{S\cdot B}}$ in terms of class intrinsic structure, and with $X_{\mathrm{T\cdot A}}$ in terms of modality information. In **cross-modality direction** $\mathrm{T(M)}$, the class structure of source data helps to uncover the latent label and structure of the missing data. In **cross-database direction** $\mathrm{T(D)}$, the complete modality information is transferred from the auxiliary database to the object database. Therefore, our knowledge transfer in two directions can mitigate the divergence between two databases and two modalities.

Specifically, in **cross-modality direction**, we set $X_{\mathrm{S}} = [X_{\mathrm{S\cdot A}}, X_{\mathrm{S\cdot B}}]$ from the same modality of two databases, $X_{\mathrm{T}} = X_{\mathrm{T\cdot A}}$ from another modality of the auxiliary database, to learn

the subspace $P_{\mathrm{T(M)}}$ from direction $\mathrm{T(M)}$ to uncover the class intrinsic information within database, while in **cross-database direction**, we set $X_{\mathrm{S}} = [X_{\mathrm{S\cdot A}}, X_{\mathrm{T\cdot A}}]$ from two modalities of auxiliary database, $X_{\mathrm{T}} = X_{\mathrm{S\cdot B}}$ from the modality of objective database, to achieve the subspace $P_{\mathrm{T(D)}}$ from direction $\mathrm{T(D)}$, transferring the modality information between databases. In detail, $P_{\mathrm{T(M)}}$ and $P_{\mathrm{T(D)}}$ are updated alternatively: first learn the projection in one direction, and then learn projection in another direction using the data embedded in the previous subspace. In our experiments, we discuss the performance of two directions transfer in different orders: $\mathrm{T(DM)}$, and $\mathrm{T(MD)}$. $\mathrm{T(DM)}$ indicates conducting cross-database transfer first and then cross-modality, while $\mathrm{T(MD)}$ conducting cross-modality transfer first. We evaluate different directions to see the best one in multi-modal databases.

3.4 Experiments

In this section, we first introduce the databases and experimental settings, and then showcase the convergence and property of the proposed M^{2}TL in two directions. We also discuss the influence of the model parameters. Finally, we compare it with several state-of-the-art transfer learning algorithms on two sets of multi-modal databases.

3.4.1 Datasets and Experiments Setting

Experiments are conducted on three sets of multimodal databases (samples shown in Fig. 3.3), which are (1) BUAA [70] and Oulu VIS-NIR face databases[2]; (2) CMU-PIE[3] and Yale B face databases[4]; The raw feature is used for those databases.

BUAA and Oulu VIS-NIR Face databases. There are 150 subjects in BUAA database and 80 subjects in Oulu database, and each has two modalities: VIS and NIR. As for BUAA, we randomly select 75 subjects with corresponding VIS images as one modality, and use the left 75 subjects with corresponding NIR images as the other modality. For Oulu, we randomly select 40 subjects with corresponding VIS images as one modality, and the left 40 subjects with corresponding NIR images as the other modality. There is no label overlap between two modalities in two databases. The size of all the images is 30×30.

[2]http://www.ee.oulu.fi/∼gyzhao/
[3]http://vasc.ri.cmu.edu/idb/html/face/
[4]http://vision.ucsd.edu/∼leekc/ExtYaleDatabase/ExtYaleB.html

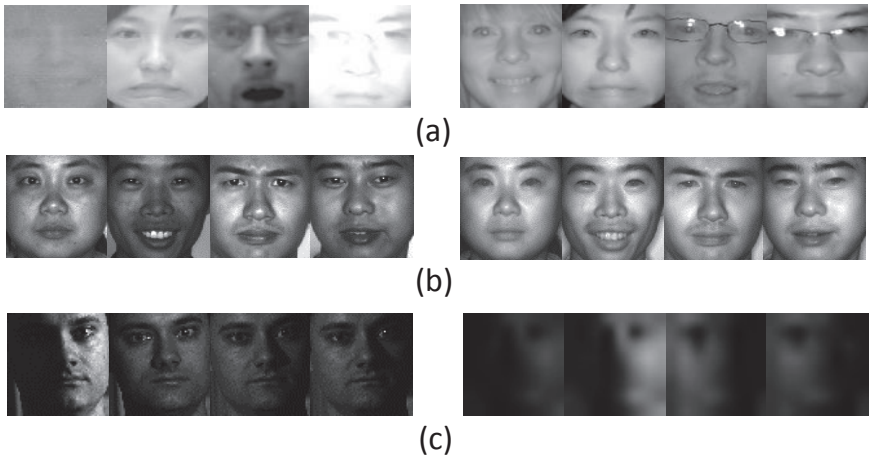
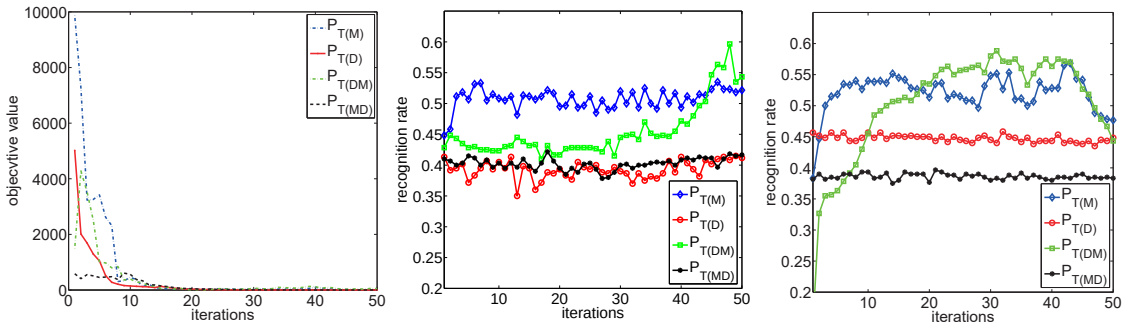


Figure 3.3: Samples from (a) Oulu NIR-VIS face database (Left: VIS image; Right: NIR image.), (b) BUAA NIR-VIS face database (Left: VIS image; Right: NIR image.), (c) CMU-PIE face database (Left: HR image; Right: LR image.)

CMU-PIE and Yale B Face databases. We focus on two different modalities: high resolution (HR) and low resolution (LR) in this experiment. We use part of CMU-PIE and Yale B databases for the experiment. For CMU-PIE with 68 subjects, the Pose C27 is used, and for Yale B with 38 subjects, the cropped images are used. We resize original images into 32×32 as HR images. While for LR images, we first downsample HR images into 8×8, then interpolate it back to 32×32, therefore, the largest dimension of the LR images is 64. Both of them are implemented by *imresize()* function in matlab. It can be observed that images of LR are very blurry from Fig. 3.3(c). Note that although there is label overlap between HR and LR in two databases, the image samples are different.

In total, we have three sets of databases: BUAA&Oulu, CMU-PIE&Yale B and each has four datasets (two modalities from two databases). So for each set of databases, we can select one dataset out of four as the test data (missing modality) and the other three as the training data. In both sets, we randomly select one sample per subject from the testing data as the reference data. Note there is no overlap between the reference and testing data. The learned projection P is applied to reduce the dimension of the testing and reference data. We repeat this five times using the nearest-neighbor as the classifier, and average results are reported. There are three sets of experiments: (1) evaluation on convergence and property in two directions; (2) influence of parameters; (3) comparisons with other transfer learning algorithms.



(a) Convergence of PCA (b) Recognition Results of PCA (c) Recognition Results of LDA

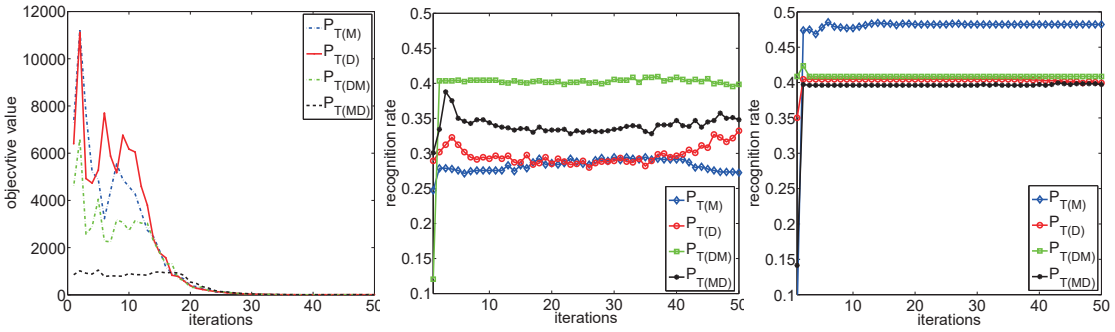
Figure 3.4: Results of convergence (a) and recognition rate (b,c) with different iterations on BUAA&Oulu database using PCA and LDA to pre-learn the low-dimensional features. The dimensions of the final subspaces are 100 for PCA and 80 for LDA. Here we only show the results of 50 iterations.

3.4.2 Convergence and Property in Two Directions

In this experiment, we first test the convergence and the recognition results over different iterations in single direction ($\mathrm{T(M)}$, $\mathrm{T(D)}$) and two directions ($\mathrm{T(DM)}$, $\mathrm{T(MD)}$). Note we only conduct one experiment for each set of databases in this subsection. Namely, we take NIR images of BUAA as the testing data for BUAA&Oulu, while HR of Yale B as the testing data for CMU-PIE&Yale B. The results of BUAA&Oulu are shown in Fig. 3.4, and that of CMU-PIE&Yale B are shown in Fig. 3.5.

Discussion: From the results, we see that our algorithms converge in different scenarios with different speeds. Besides, we observe that cross-modality direction plays a key role in *Missing Modality Problem*. Another observation is that good results can be achieved in a few iterations. We believe it is because more iterations does not necessarily benefit more, and may even incur negative transfer. Therefore, we set maximum iteration as 30 in the following experiment.

Specifically, for BUAA&Oulu (Fig. 3.4), the best performance is achieved by $\mathrm{P_{T(DM)}}$, meaning knowledge transfer in two directions: first cross-database then cross-modality. This shows (1) our transfer in two directions can help improve the performance, and (2) the order of two directions is very important. In addition, we find $\mathrm{P_{T(DM)}}$ and $\mathrm{P_{T(D)}}$ achieve similar results. The reasons might be that the similarity between BUAA and Oulu is very low, namely, different capture devices, detailed lighting conditions, subject identities. Therefore, $\mathrm{P_{T(D)}}$ can transfer less information to the missing modality than $\mathrm{P_{T(M)}}$. Comparing $\mathrm{P_{T(M)}}$ with $\mathrm{P_{T(DM)}}$, the later one still works better. We believe



(a) Convergence of PCA (b) Recognition Results of PCA (c) Recognition Results of LDA

Figure 3.5: Results of convergence (a) and recognition rate (b,c) with different iterations on CMU-PIE&Yale B database using PCA and LDA to pre-learn the low-dimensional features. The dimensions of the final subspaces are 60 for PCA and LDA. Here we only show the results of 50 iterations.

that the cross-database transfer still helps. Note we set a higher dimension (here 500) in the first direction, which helps preserve lots of original information.

For CMU-PIE&Yale B (Fig. 3.5), we find that the largest subspace dimension of LR modality is 64, due to downsampling process to the size of 8×8. Therefore, we first set the subspace dimension of the first direction as 64, then learn the second subspace with dimension 60. In PCA, $\mathrm{P_{T(DM)}}$ performs better than models in one direction. In LDA case, however, $\mathrm{P_{T(M)}}$ achieves better results. This is because, in PCA case, $\mathrm{P_{T(D)}}$ helps to mitigate the divergence between two databases in terms of data distribution and transfer more modality information to the objective database, while in LDA case, the label information in the auxiliary database may not be applicable to the object database. This becomes significant when the number of classes in two databases are different, which is similar to our previous results in [15].

In addition, we find that two directions' models show reasonably good results at the first round and then gradually and slightly increase after this. Therefore, we believe if we fine tune the parameters of $\mathrm{P_{T(M)}}$ and $\mathrm{P_{T(D)}}$ in the first round, it may already adequate to output comparable results. Moreover, since our method is still in the line of traditional transfer learning, one round with two directional transfer is equal to the whole process of traditional transfer learning methods. Consequently, we compare one iteration results in the following comparison experiments.

3.4.3 Recognition Results

In the second set of experiments, we compared our method with TSL [71], LTSL [16], RDALR [82], GFK [9], DASA [12] and our conference version L^{2}STL [15] in different subspace settings: PCA[64], LDA [65], Unsupervised LPP (ULPP) and Supervised LPP (SLPP) [66] for BUAA&Oulu and CMU-PIE&Yale B. Whilst we only evaluate PCA subspace for ALOI-100&COIL-100. Since the latter three ones are domain adaptation algorithms, assuming the label sets of source and target domains are identical, they usually use transformed source data to predict unknown target data. However, in our *Missing Modality Problem*, label sets of source and target domains are different, so we follow their original settings except that we use one reference image per subject for the test, like our NN classifier. Specifically, for RDALR, we first learn the rotation W on source, and then combine the rotated source and target to train the subspace for extracting features in testing stage. For GFK and DASA, we first learn kernel mapping G or subspace alignment M from source and target data using different subspace learning methods, and then apply the learned matrix to testing data. TSL, LTSL, L^{2}STL and our method work in the same way, by learning subspace projection matrices from source and target data in the training stage, and then apply to missing modality in the testing step.

For comparison methods, we set the source data as $[X_{\mathrm{S\cdot A}}, X_{\mathrm{T\cdot A}}]$ and target domain as $X_{\mathrm{S\cdot B}}$. For our method, we show the best results by comparing $\mathrm{P_{T(M)}}$, $\mathrm{P_{T(D)}}$, $\mathrm{P_{T(DM)}}$ and $\mathrm{P_{T(MD)}}$. Tables I, II, III show the average results with standard deviations of 4 cases by changing training and testing data settings. Fig. 3.6, 3.7 show the results in different dimensions for one case.

Discussion: It can be seen that our method performs better than comparison algorithms. Both LTSL and RDALR perform better than TSL, which demonstrates that low-rank constraint is helpful on data alignment. Compared to one direction knowledge transfer, e.g., LTSL and RDALR, the proposed method works better. One reason is our method can compensate for missing modality through the auxiliary database, which is also helpful in knowledge transfer between modalities in the same database. In supervised cases of CMU-PIE&Yale B, our method only learns the subspace in one direction between modalities, but still achieves good performance. We attribute this to the latent factor from the source data which uncovers the missing part of testing data.

As we proposed, the introduction of pre-learned low-dimensional feature stabilizes the optimization compared with our previous work [15] and LTSL [16], both of which learn the projection on two sides of the low-rank constraint. Our current model converges well with appealing results, especially when we set a high dimensionality for the first subspace projection. However, in our

Table 3.1: Average recognition rates (%) with standard deviations of all compared methods on BUAA&Oulu face database, where the test data, respectively, are NIR of BUAA (**Case 1**), VIS of BUAA (**Case 2**), NIR of Oulu (**Case 3**) and VIS of Oulu (**Case 4**). We show the best results of our proposed four algorithms: T(M), T(D), T(MD) and T(DM). Red color denotes the best recognition rates. Blue color denotes the second best recognition rates.

Methods		TSL [71]	RDALR [82]	GFK [9]	LTSL [16]	DASA [12]	L^2TSL [15]	Ours
Case 1	PCA	35.82±0.76	40.21±0.67	38.34±0.83	47.21±0.54	59.43±0.62	52.32±0.67	59.02±0.46
	LDA	31.31±0.32	38.52±0.52	12.7±0.12	42.38±0.43	11.59±0.10	48.72±0.42	62.48±0.30
	ULPP	29.28±0.45	42.84±0.37	40.21±0.25	50.81±0.85	41.31±0.83	59.68±0.48	58.32±0.65
	SLPP	36.86±0.38	47.27±0.42	39.56±0.36	53.57±0.52	18.17±0.15	63.71±0.62	66.68±0.35
Case 2	PCA	37.06±0.34	33.76±0.39	42.39±0.49	38.39±0.46	38.37±0.49	49.79±0.52	51.21±0.56
	LDA	28.39±0.12	34.57±0.23	15.84±0.18	41.38±0.38	18.76±0.07	43.23±0.36	52.98±0.25
	ULPP	38.29±0.31	39.88±0.42	39.29±0.23	41.28±0.35	37.48±0.51	49.34±0.35	55.71±0.48
	SLPP	46.88±0.51	50.28±0.28	48.39±0.39	56.79±0.53	34.76±0.22	60.73±0.58	61.54±0.63
Case 3	PCA	39.26±0.23	41.37±0.25	39.59±0.38	41.89±0.33	42.25±0.10	48.34±0.43	50.21±0.26
	LDA	42.25±0.51	36.58±0.24	26.87±0.38	50.76±0.63	23.83±0.29	56.82±0.42	64.77±0.54
	ULPP	47.37±0.43	42.39±0.62	28.38±0.35	48.24±0.32	52.58±0.11	50.83±0.42	58.89±0.43
	SLPP	45.75±0.38	48.28±0.41	45.38±0.47	54.78±0.52	41.08±0.06	55.71±0.32	64.58±0.55
Case 4	PCA	31.59±0.54	39.77±0.62	39.29±0.71	43.35±0.58	48.00±0.20	46.32±0.48	51.25±0.43
	LDA	40.34±0.42	42.38±0.33	38.36±0.51	48.28±.35	29.03± 0.71	67.54±0.34	74.47±0.76
	ULPP	39.26±0.51	47.57±0.35	42.89±0.72	52.38±0.53	56.50±0.09	58.23±0.32	61.91±0.20
	SLPP	36.25±0.24	49.39±0.29	29.38±0.35	58.89±0.25	38.50±0.11	68.54±0.32	73.31±0.39

previous work [15], if we set a high dimensionality at the beginning for better performance, it may stop within 5 iterations due to unstable projections. Compared with our previous work [15], our current one can achieve better results in BUAA&Oulu database as we can set a higher dimension in the first direction (e.g. 500), while we can only set 64 as the largest dimension in CMU-PIE&Yale B, as we mentioned before. So in CMU-PIE &Yale B, our previous conference work [15] can achieve better results in most cases. One reason is learning projection on two sides of low-rank constraint can make it more flexible to uncover information from two domains. On the other hand, we introduce pre-learned low-dimensional feature into latent low-rank constraint to make our model more stable and robust. However, such pre-learned feature may bring in some noises. For these reasons, we can see our revised method in journal extension, M^2TL, does not always outperform our previous method

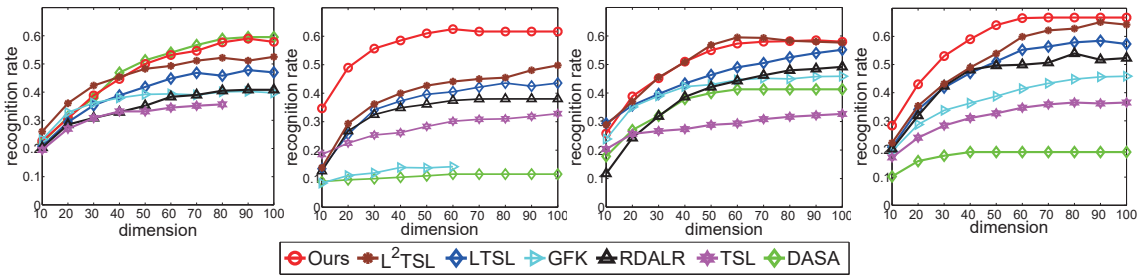


Figure 3.6: Results of six algorithms on BUAA&Oulu face database (**Case 1**) in four different subspaces. Subspace methods from left to right are PCA, LDA, ULPP and SLPP. We show the best results of our proposed four algorithms: T(M), T(D), T(MD) and T(DM).

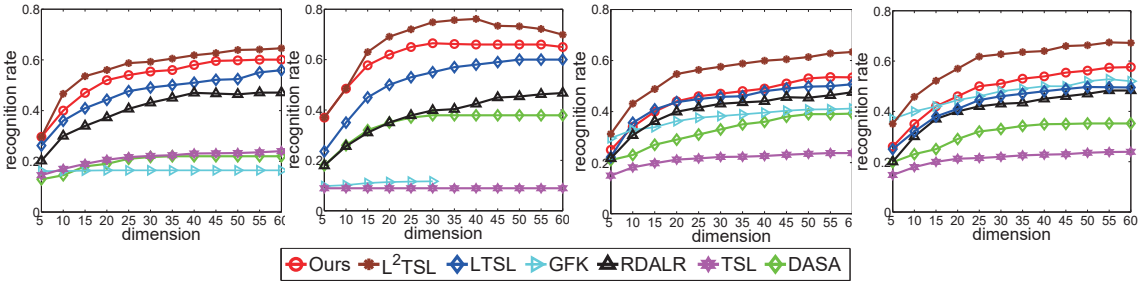


Figure 3.7: Results of six algorithms on CMU-PIE&Yale B face database (**Case 1**) in four different subspaces. Subspace methods from left to right are: PCA, LDA, ULPP and SLPP. We show the best results of our proposed four algorithms: T(M), T(D), T(MD) and T(DM).

in conference version, L^{2}TSL, as seen in Table II.

3.4.4 Parameter Property & Training Time

In this section, we evaluate the following parameters: λ, α and β. We test them one by one by keeping other parameters fixed. We take NIR of BUAA as the testing and others as the training data, by applying PCA to pre-learn the low-dimensional subspace.

We can see from the results shown in Fig. 3.8 that the best performance usually appears at small values (e.g. 0.001) for λ, α and β. We also show the results when one of three parameters is removed, meaning one of λ, α and β is set as zero. It is clear that all of them are helpful on improving the performance.

We evaluate the computational cost of different methods (GFK [9], LTSL [16], L^{2}TSL [15] and Ours) in PCA situation. Taking BUAA&Oulu database as an example, we use the **Case 1** and run 100 iterations for LTSL[95], L^{2}TSL [15] and Ours. We experiment on Matlab 2014 with CPU i7-3770 and memory size (32 GB). Table IV shows the training time, whose unit is *second*.

Table 3.2: Average recognition rates (%) with standard deviations of all compared methods CMU-PIE&Yale B face database , where the test data, respectively, are HR of CMU-PIE (**Case 1**), LR of CMU-PIE (**Case 2**), HR of Yale B (**Case 3**) and LR of Yale B (**Case 4**). We show the best results of our proposed four algorithms: T(M), T(D), T(MD) and T(DM). Red color denotes the best recognition rates. Blue color denotes the second best recognition rates.

Methods		TSL [71]	RDALR [82]	GFK [9]	LTSL [16]	DASA [12]	L^2TSL [15]	Ours
Case 1	PCA	22.06±0.30	42.14±0.44	17.32±0.23	56.36±0.33	22.22±0.39	60.82±0.36	60.05±0.43
	LDA	09.12±0.00	42.85±0.49	12.33±0.28	60.15±0.63	37.51±0.32	74.46±0.65	66.53±0.41
	ULPP	22.26±0.19	44.56±0.46	40.23±0.45	49.23±0.35	39.60±0.29	59.47±0.59	53.47±0.34
	SLPP	22.85±0.32	48.35±0.55	42.85±0.65	49.75±0.30	35.99±0.52	62.56±0.93	57.66±0.69
Case 2	PCA	20.36±0.37	42.83±0.61	17.33±0.15	47.83±0.38	20.51±0.18	53.24±0.73	50.64±0.65
	LDA	50.85±0.73	47.84±0.55	24.13±0.32	54.54±0.42	45.58±0.34	60.28±0.62	58.14±0.59
	ULPP	27.46±0.26	50.15±0.49	23.44±0.42	56.73±0.52	48.53±0.29	58.36±0.63	57.84±0.63
	SLPP	48.75±0.37	47.35±0.35	49.84±0.46	53.2±0.45	49.19±0.34	54.54±0.49	58.55±0.55
Case 3	PCA	25.44±0.27	38.34±0.45	08.32±0.00	40.44±0.55	24.32±0.25	41.34±0.43	41.86±0.28
	LDA	08.26±0.00	38.94±0.19	11.23±0.21	43.23±0.28	19.81±0.22	45.14±0.45	48.26±0.35
	ULPP	35.35±0.61	38.56±0.53	40.76±0.39	39.35±0.49	29.14±0.25	42.23±0.43	43.35±0.32
	SLPP	35.54±0.37	37.43±0.45	37.85±0.42	38.46±0.52	23.43±0.28	43.44±0.76	51.02±0.60
Case 4	PCA	20.05±0.26	32.34±0.42	08.33±0.02	32.13±0.36	21.59±0.18	38.43±0.64	34.64±0.51
	LDA	21.35±0.31	32.95±0.37	27.81±0.19	35.65±0.30	33.12±0.29	37.84±0.56	44.53±0.34
	ULPP	15.22±0.13	35.16±0.27	33.36±0.34	37.83±0.45	32.49±0.43	41.63±0.54	38.46±0.44
	SLPP	20.35±0.13	37.85±0.34	32.25±0.26	36.74±0.34	33.45±0.32	41.32±0.34	44.66±0.28

From the results, we can observe our algorithm works more efficiently than LTSL, especially than L^2TSL. We attribute to an efficient optimization solution designed to our problem, which avoids the time-consuming matrix multiplication and inverse due to the introduced relaxing variables.

Table 3.3: Training time (*second*) of four algorithms on **Case 1** of BUAA&Oulu face database

Methods	GFK [9]	LTSL [16]	L^2TSL [15]	Ours
Training Time	2.83	300.55	1305.32	254.52

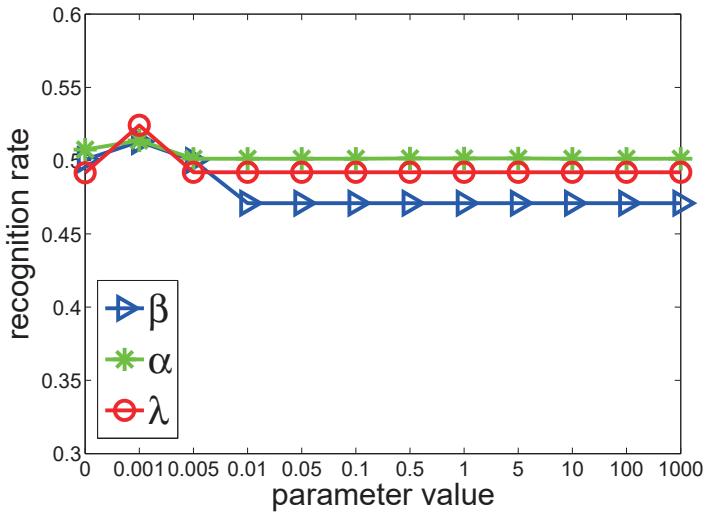


Figure 3.8: Recognition results of different values for three parameters α, β, and λ. We evaluate the influence of each parameter by fixing others.

3.5 Conclusion

In this chapter, we proposed a novel Latent Low-rank Transfer Subspace Learning algorithm for *Missing Modality Problem*, named as M^{2}TL. With the auxiliary database, our proposed algorithm is capable of transferring knowledge in two directions: cross-modality within database and cross-database. By introducing a latent low-rank constraint, our algorithm can learn appropriate subspaces to better recover the missing information of the testing modality from two directions. Experiments on three sets of multi-modal databases, involving face and object data, have shown that our method can better tackle the *Missing Modality Problem* in knowledge transfer, compared with several existing transfer learning methods.

Chapter 4

Deep Feature Learning for Face Recognition

4.1 Background

In the recent years, deep learning has attracted considerable interests in computer vision field, as it has achieved promising performance in various tasks, e.g., image classification [96], object detection [97] and face recognition [98]. Generally, deep structure learning tends to extract hierarchical feature representations directly from raw data. Recent representative research works include: deep convolutional neural networks [99], deep neural networks [100], deep auto-encoder [101], and deeply-supervised nets [102].

Among different deep structures, auto-encoder (AE) [23] has been treated as robust feature extractors or pre-training scheme in various tasks [24, 25, 26, 27, 28, 29]. Conventional AE was proposed to encourage similar or identical input-output pairs where the reconstruction loss is minimized after decoding [23]. Follow-up work with various additive noises in the input layer is able to progressively purify the data, which fulfills the purpose "denoising" against unknown corruptions in the testing data [30]. These works as well as the most recent AE variants, e.g., multi-view AE [28] and bi-shift AE [26], all assume the training data are clean, but can be intentionally corrupted. In fact, real-world data subject to corruptions such as changing illuminations, pose variations, or self-corruption do not meet the assumption above. Therefore, learning deep features from real-world corrupted data instead of intentionally corrupted data with additive noises becomes critical to build robust feature extractor that is generalized well to corrupted testing data. To the best of our

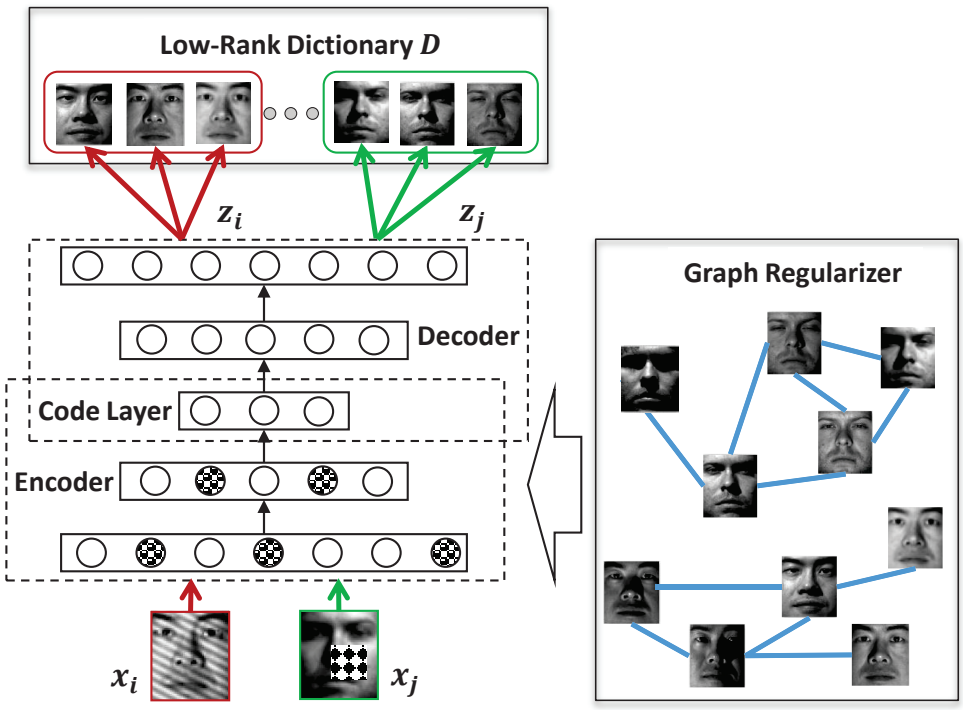


Figure 4.1: Illustration of our proposed algorithm. Corrupted data x_i, x_j are the inputs of the deep AE. After encoding and decoding process, the reconstructed x_i, x_j are encouraged to be close to Dz_i, Dz_j on the top, where D is the learned clean low-rank dictionary and z_i, z_j are corresponding coefficients. In addition, graph regularizers are added to the encoder layers to pass on the locality information.

knowledge, such AE based deep learning scheme has not been discussed before.

Recently, low-rank matrix constraint has been proposed to learn robust features from corrupted data. Specifically, when data are lying in a single subspace, robust PCA (RPCA) [103] could well recover the corrupted data by seeking a low-rank basis. While low-rank representation (LRR) [104] is designed to recover corrupted data and rule out noises in case of multiple subspaces. Due to these technical merits, low-rank modeling has already been successfully used in different scenarios, e.g., multi-view learning [88], transfer learning [105, 14, 106], and dictionary learning [107]. However, fewer works link the low-rank modeling to deep learning framework for robust feature learning.

Inspired by the above facts, we develop a novel algorithm named as Deep Robust Encoder (DRE) with locality preserving low-rank dictionary. The core idea is to jointly optimize deep AE and a clean low-rank dictionary, which can rule out noises and extract robust deep features in a unified

framework (Figure 5.2). To sum up, our contributions are three folds as follows:

- A low-rank dictionary and deep AE are jointly optimized based on the corrupted data, which can progressively denoise the already corrupted features in the hidden layers so that robust deep AE could be achieved for corrupted testing data.
- The newly designed loss function, which is based on the clean low-rank dictionary and preserved locality information in the output layer, penalizes the corruptions or distortions, meanwhile ensures that the reconstruction is noise free.
- Graph regularizers are developed to guide feature learning in each encoding layer to preserve more geometric structures within the data, in either unsupervised or supervised fashions.

4.2 The Proposed Algorithm

In this section, we first introduce our motivation, and then propose our deep robust encoder through locality preserving low-rank dictionary. Finally, we present an efficient solution to the proposed framework.

4.2.1 Motivation

Intentional corruptions, e.g., random noises are added artificially while real-world ones are from data itself, e.g., varied lightings or occlusion. Most existing AE and its variants, e.g., DAE, take advantage of different additive noises on the clean data to improve the robustness of deep models. During the deep encoding/decoding process, the perturbed input data are gradually recovered. In this way, the learned deep model is able to tolerate certain corruptions simulated by the additive noises.

However, this raises two problems. First, the robustness of the system completely relies on the formulations of the noises. The richer the noisy patterns are, the better the performance will be. This inevitably increases the computational burden. In the worst case, the learned deep structure may not be well generalized to the unseen testing data. Second, real-world data usually suffer from contaminations of varied sources, and building robust feature extractors to rule out existing noises is more reasonable. In addition, recent advances in low-rank matrix modeling cast a light on denoising for data that are already corrupted. Based on these observations, we propose to jointly learn a deep AE framework and a clean low-rank dictionary to actively mitigate the noises or corruptions within the data (Figure 5.4).

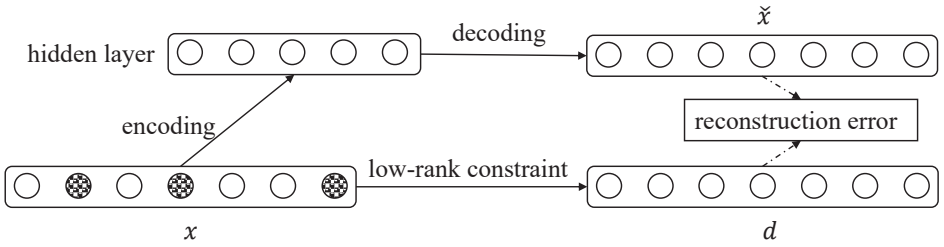


Figure 4.2: The AE architecture with low-rank dictionary. A corrupted sample x is correlated to a low-rank clean version d. The AE then maps it to hidden layer (via encoder layer) and attempts to reconstruct x via decoder layer, generating reconstruction $\check{x}$. Finally, reconstruction error can be measured by different loss functions.

4.2.2 Locality Preserving Low-rank Dictionary Learning

Suppose training data $X \in \mathbb{R}^{d\times n}$ has n samples and $x_i \in \mathbb{R}^d$ represents the i-th sample. For AE with single hidden layer [23, 100], it is usually consisted of two parts, encoder and decoder. The encoder, denoted as f_1, attempts to map the input x_i into hidden representations, while the decoder, denoted as f_2, tries to map the hidden representation back to the input x_i. A typical cost function with square loss for AE can be formulated as:

$$\min_{W_1,b_1,W_2,b_2} \sum_{i=1}^{n} \left\|x_i - f_2(f_1(x_i))\right\|_2^2, \tag{4.1}$$

where $\{W_1 \in \mathbb{R}^{r\times d}, b_1 \in \mathbb{R}^r\}, \{W_2 \in \mathbb{R}^{d\times r}, b_2 \in \mathbb{R}^d\}$ are the parameters for encoding and decoding, respectively. Specifically, we have $f_1(x_i) = \varphi(W_1x_i + b_1)$ and $f_2(f_1(x_i)) = \varphi(W_2f_1(x_i) + b_2)$, where $\varphi(\cdot)$ is an element-wise "activation function", which is usually nonlinear, such as sigmoid function or tanh function. DAE manually involves artificial noise into the input training data so that it aims to train a denoising auto-encoder to remove the random noise.

In reality, however, x_i is usually corrupted already due to environmental factors or noises from the collecting devices. Intuitively, we need to build a network by detecting and removing noise from the corrupted data so that it could better generalize to corrupted testing data. To this end, we propose our robust auto-encoder with low-rank dictionary learning:

$$\min_{W_1,b_1,W_2,b_2,D} \sum_{i=1}^{n} \|d_i - f_2(f_1(x_i))\|_2^2 + \lambda \mathrm{rank}(D), \tag{4.2}$$

where $d_i \in \mathbb{R}^d$ is the i-th column of low-rank $D \in \mathbb{R}^{d\times n}$ and λ is the tradeoff parameter. $\mathrm{rank}(\cdot)$ means the rank operator of a matrix, which encourages to build a clean and compact basis. Generally,

the convex surrogate of rank problem, i.e., nuclear norm $\|\cdot\|_*$ will be employed to solve the rank minimization problem [103].

However, similar to the conventional AE and its variants, the point-to-point reconstruction scheme in Eq. (4.2) only considers one-to-one mapping, which may overfit the data and skip the structure knowledge within the data. To that end, we propose a novel locality preserving low-rank dictionary learning by introducing a new coefficient vector z_i to maintain the locality of each sample x_i throughout the network:

$$\min_{W_1,b_1,W_2,b_2,D} \sum_{i=1}^{n} \|Dz_i - f_2(f_1(x_i))\|_2^2 + \lambda\|D\|_*, \tag{4.3}$$

where $z_i \in \mathbb{R}^n$ is the coefficient vector for sample x_i w.r.t. dictionary D. There are different strategies to obtain the coefficient vector z_i, in either unsupervised or supervised fashion, depending on the availability of label information. Specifically, the j-th element in z_i is defined as:

$$z_{ij} = \begin{cases} 1, & \text{if } i = j, \\ \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), & \text{if } x_i \in \mathcal{N}_{k_1}(x_j), \\ 0, & \text{otherwise}, \end{cases} \tag{4.4}$$

where $x_i \in \mathcal{N}_{k_1}(x_j)$ means x_i is within the k_1 nearest neighbors of x_j. Specifically, we could define the locality-preserving coefficients z_i in two fashions. For unsupervised case, the k_1 nearest neighbors are searched from the whole data, while for supervised case, the k_1 nearest neighbors are searched from the data within the same class to x_i. Actually, we could easily extend semi-supervised scenario. Note σ is a bandwidth for Gaussian kernel (we set $\sigma = 5$ in this paper).

To sum up, our regularized deep auto-encoder transform the original AE's point-to-point reconstruction strategy to our point-to-set reconstruction so that we could preserve more discriminative information. To further guide the locality preserving dictionary learning in the output layer, we propose to couple the discriminant graph regularizers with hidden feature learning during the optimization:

$$\begin{aligned} \min_{W_1,b_1,W_2,b_2,D} & \sum_{i=1}^{n} \|Dz_i - f_2(f_1(x_i))\|_2^2 + \lambda\|D\|_* \\ & +\alpha \sum_{j=1}^{n}\sum_{k=1}^{n} s_{jk}(f_1(x_j) - f_1(x_k))^2, \end{aligned} \tag{4.5}$$

where s_{jk} is the similarity between x_j and x_k. α is the balance parameter.

Specifically, s_{jk} can be calculated in unsupervised and supervised fashions as well:

$$s_{jk} = \begin{cases} \exp\left(-\frac{\|x_j - x_k\|^2}{2\sigma^2}\right), & \text{if } x_j \in \mathcal{N}_{k_2}(x_k), \\ 0, & \text{otherwise}, \end{cases} \tag{4.6}$$

where $x_j \in \mathcal{N}_{k_2}(x_k)$ means x_j is within the k_2 nearest neighbors of x_k. In the same way as z_i, the k_2 nearest neighbors are selected from the whole dataset for unsupervised case, while the k_2 nearest neighbors are selected from the data within the same class to x_j for supervised case.

4.2.3 Deep Architecture

Considering the learning objective in Eq. (4.5) as a basic building block, we can train a more discriminant deep model. Existing popular training schemes for deep auto-encoder includes Stacked Auto-Encoder (SAE) [30] and Deep Auto-Encoder [101]. However, as our learning objective/building block is different from theirs, we have a different training scheme for the deep structure.

Assume we have L encoding layers and L decoding layers in our deep structure which minimizes the following loss:

$$\min_{W_l, b_l, D} \sum_{i=1}^{n} \|Dz_i - \bar{x}_i\|_2^2 + \lambda\|D\|_* + \alpha \sum_{l=1}^{L} \sum_{j=1}^{n} \sum_{k=1}^{n} s_{jk}(f_l(x_j) - f_l(x_k))^2, \tag{4.7}$$

where $\bar{x}_i$ is the output with a series of encoding and decoding from the input x_i. $\{W_l, b_l\}, (1 \leq l \leq L)$ are the encoding parameters while $\{W_l, b_l\}, (L+1 \leq l \leq 2L)$ are the decoding parameters. The third term sums up the graph regularizers from each encoding layer to guide the locality preserving low-rank dictionary learning in the output layer.

4.2.4 Optimization

Eq.(4.7) is difficult to address because of the non-convexity and non-linearity of the building block formulated in Eq. (4.5). To this end, we develop an alternating solution to iteratively update the encoding & decoding functions $f_l(1 \leq l \leq 2L)$ and dictionary D. First we list the low-rank dictionary learning, then provide the regularized deep auto-encoder optimization.

4.2.4.1 Low-rank Dictionary Learning

When $f_l(1 \leq l \leq 2L)$ are fixed, the objective function in Eq.(4.7) degenerates to a conventional low-rank recovery problem, which can be solved by augmented Lagrange multiplier

algorithm [108]. To that end, we first involve a relaxing variable J, and write down its equivalent formulation as:

$$\min_{D,J} \|\bar{X} - DZ\|_{\mathrm{F}}^2 + \lambda\|J\|_*, \ \text{s.t.} \ D = J,$$

where $\bar{X} = [\bar{x}_1, \cdots, \bar{x}_n]$ and $Z = [z_1, \cdots, z_n]$. $\|\cdot\|_{\mathrm{F}}^2$ is Frobenius norm of a matrix. Then we derive the corresponding augmented Lagrangian function w.r.t. D, J:

$$\|\bar{X} - DZ\|_{\mathrm{F}}^2 + \lambda\|J\|_* + \langle R, D - J\rangle + \frac{\mu}{2}\|D - J\|_{\mathrm{F}}^2,$$

where R is the Lagrange multiplier and $\mu > 0$ is the penalty parameter. $\langle , \rangle$ is the matrix inner product operator. Specifically, we have the following updating rules for D, J one variable at time t:

$$J_{t+1} = \arg\min_{J} \frac{\lambda}{\mu_t}\|J\|_* + \frac{1}{2}\|J - D_t - \frac{R_t}{\mu_t}\|_{\mathrm{F}}^2, \tag{4.8}$$

which can be effectively addressed by the singular value thresholding (SVT) operator [60].

$$\begin{aligned} D_{t+1} = & \ \arg\min_{D} \|\bar{X} - DZ\|_{\mathrm{F}}^2 + \langle R_t, D - J_{t+1}\rangle + \tfrac{\mu_t}{2}\|D - J_{t+1}\|_{\mathrm{F}}^2 \\ = & \ (2\bar{X}Z^\top + \mu_t J_{t+1} - R_t)(2ZZ^\top + \mu_t \mathrm{I}_n)^{-1}, \end{aligned} \tag{4.9}$$

where $\mathrm{I}_n \in \mathbb{R}^{n\times n}$ is an identical matrix.

4.2.4.2 Deep Robust Encoder Learning

When D is fixed, the objective function in Eq.(4.7) can be reformulated to minimize the following objective function:

$$\mathcal{L} = \sum_{i=1}^{n} \|\bar{x}_i - \bar{d}_i\|_2^2 + \alpha \sum_{l=1}^{L}\sum_{j=1}^{n}\sum_{k=1}^{n} s_{jk}(f_l(x_j) - f_l(x_k))^2,$$

where $\bar{d}_i = Dz_i$. Since the loss function is smooth and twice-differentiable, we can still adopt L-BFGS optimizer [109] to deal with this unconstrained problem, whose updating rules at time t are shown as follows:

$$\begin{cases} W_{l,t+1} = W_{l,t} - \eta_t H_{l,t} \dfrac{\partial \mathcal{L}}{\partial W_l}|_{W_{l,t}}, \\ b_{l,t+1} = b_{l,t} - \eta_t G_{l,t} \dfrac{\partial \mathcal{L}}{\partial b_l}|_{b_{l,t}}, \end{cases} \tag{4.10}$$

in which η_t denotes the learning rate, $H_{l,t}$ and $G_{l,t}$ are the approximations for the inverse Hessian matrices of $\mathcal{L}$ w.r.t. to W_l and b_l, respectively. The detailed formulations and discussions of η_t, $H_{l,t}$

and $G_{l,t}$ are trivial, which can be referred to [109]. In this section, we mainly focus on the derivatives of $\mathcal{L}$ w.r.t. to W_l and b_l.

For the **decoding layers** $(L+1 \leq l \leq 2L)$, we have:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \sum_{i=1}^{n} \mathcal{F}_{i,l} \mathbf{f}_{i,l-1}^{\top}, \quad \frac{\partial \mathcal{L}}{\partial b_l} = \sum_{i=1}^{n} \mathcal{F}_{i,l},$$

where $\mathbf{f}_{i,l-1} = f_{l-1}(x_i)$ is the l-1^{th}-layer hidden layer feature and the updating equations are computed as follows:

$$\mathcal{F}_{i,2L} = 2(\bar{x}_i - \bar{d}_i) \odot \varphi'(\mathbf{u}_{i,2L}),$$

$$\mathcal{F}_{i,l} = (W_{l+1}^{\top} \mathcal{F}_{i,l+1}) \odot \varphi'(\mathbf{u}_{i,l}).$$

Here the operator $\odot$ denotes the element-wise multiplication, and $\mathbf{u}_{i,l}$ is computed by $\mathbf{u}_{i,l} = W_l \mathbf{f}_{i,l-1} + b_l$.

For the **encoding layers** $(1 \leq l \leq L)$, we have:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W_l} &= \sum_{i=1}^{n} \mathcal{F}_{i,l} \mathbf{f}_{i,l-1}^{\top} + \\
&\qquad 2\alpha \sum_{p=l}^{L} \sum_{j=1}^{n} \sum_{k=1}^{n} s_{jk} (\mathcal{G}_{jk,p} \mathbf{f}_{j,p-1}^{\top} + \mathcal{G}_{kj,p} \mathbf{f}_{k,p-1}^{\top}), \\
\frac{\partial \mathcal{L}}{\partial b_l} &= \sum_{i=1}^{n} \mathcal{F}_{i,l} + 2\alpha \sum_{p=l}^{L} \sum_{j=1}^{n} \sum_{k=1}^{n} s_{jk} (\mathcal{G}_{jk,p} + \mathcal{G}_{kj,p}),
\end{aligned}$$

in which $\mathcal{G}_{jk,l}$ and $\mathcal{G}_{kj,l}$ are calculated as follows:

$$\mathcal{G}_{jk,L} = (\mathbf{f}_{j,l} - \mathbf{f}_{k,l}) \odot \varphi'(\mathbf{u}_{j,L}),$$

$$\mathcal{G}_{kj,L} = (\mathbf{f}_{k,l} - \mathbf{f}_{j,l}) \odot \varphi'(\mathbf{u}_{k,L}),$$

$$\mathcal{G}_{jk,l} = (W_{l+1}^{\top} \mathcal{G}_{jk,l+1}) \odot \varphi'(\mathbf{u}_{j,l}),$$

$$\mathcal{G}_{kj,l} = (W_{l+1}^{\top} \mathcal{G}_{kj,l+1}) \odot \varphi'(\mathbf{u}_{k,l}).$$

To that end, we can optimize low-rank dictionary and deep auto-encoder iteratively until convergence. The entire procedure of two sub-problems is listed in **Algorithm 1**. Before the

Algorithm 1: Solution to Problem (4.7)

Input: $\{X, y\}$, α, λ, $\eta_0 = 0.2, \varepsilon = 10^{-6}, t = 0$,
$\mu_0 = 10^{-6}, \rho = 1.3, \mu_{\max} = 10^6$, and $t_{\max} = 10^3$.

while not converged **or** $t < t_{\max}$ **do**

Step 1. Update low-rank dictionary via (4.8),(4.9);

Step 2. Update the deep auto-encoder:

for $l = 2L, \cdots, 1$ **do**

Compute derivatives $\frac{\partial \mathcal{L}}{\partial W_l}, \frac{\partial \mathcal{L}}{\partial b_l}$;

end

for $l = 1, \cdots, 2L$ **do**

Update W_l, b_l using (4.10);

end

Step 3. Update parameters:

$R_{t+1} = R_t + \mu_t(D_{t+1} - J_{t+1})$; $\eta_{t+1} = 0.95 \times \eta_t$;
$\mu_{t+1} = \min(\mu_{\max}, \rho\mu_t)$; $t = t + 1$.

Step 4. Check convergence:

$|\mathcal{L}_{t+1} - \mathcal{L}_t| < \varepsilon$, $\|D_{t+1} - J_{t+1}\|_\infty < \varepsilon$.

end

Output: $\{W_l, b_l, D, J\}$.

alternative updating, the network parameters $f_l (1 \leq l \leq 2L)$ are initialized through deep auto-encoder with the input and the target as X [101], whilst D is directly set as original data X for initialization.

4.3 Experiments

In this section, we conduct experiments to systematically evaluate our algorithm. First, we present the details of datasets and experimental settings. Then we do self-evaluation on our algorithm and present the comparison results with several state-of-the-art algorithms. Finally, we further testify several properties of the proposed algorithm, e.g., impacts of layer size, parameter analysis.

4.3.1 Datasets & Experimental Settings

CMU-PIE Face dataset[1] contains 68 subjects under different poses subject to large appearance differences. In addition, for each pose, there are 21 various illumination conditions. We use

[1] http://vasc.ri.cmu.edu/idb/html/face/

Table 4.1: Recognition results (%) of 4 approaches on different setting of three datasets.

	PIE-1	PIE-2	PIE-1c	PIE-2c	ALOI-c
AE	83.58±0.11	82.79±0.13	74.95±0.14	73.89±0.12	80.98±0.98
LAE	85.87±0.16	85.08±0.14	77.82±0.12	76.14±1.45	82.84±1.26
L^2AE-u	86.98±0.09	86.45±0.11	79.23±0.11	79.02±0.12	83.42±0.87
L^2AE-s	87.67±0.10	87.54±0.12	80.14±0.10	79.96±0.11	86.27±0.75

face images from 8 different poses to construct various evaluation sets. The sizes of them vary from 2 to 5. Basically, we randomly select 15 images per pose per subject to build the training set while the left as the testing set. The face images are cropped and resized to 64×64, and the raw features are used as the inputs.

Note that previous algorithms, e.g., DAE [30], adopted the "corrupted" data with random noise as the input for training while using the "original" data for testing. However, we assume the data are "already corrupted" and we manage to detect and remove the noise. Thus, we adopt the "same" types of training and testing data without intentional corruptions. Notably, to challenge all comparisons, we introduce additional noises to the datasets that have already been corrupted by poor lighting or arbitrary views. Such practice can be found in previous work [107].

4.3.2 Self-Evaluation

In this section, we mainly testify if our low-rank dictionary D and locality preserving term $Z = [z_1, \cdots, z_n]$ would facilitate our robust feature learning. Specifically, we define the deep version of Eq.(4.2) as LAE (Auto-encoder with low-rank dictionary) and deep version of Eq.(4.3) as L^2AE (Auto-encoder with locality preserving low-rank dictionary). For L^2AE, we have two ways to learn Z, that is, we set $k_1 = k_2 = 5$ for all cases in unsupervised fashion (L^2AE-u), while we set k_1, k_2 as the size of each class for supervised fashion (L^2AE-s). A four-layer scheme is applied for all the comparisons for simplicity. We adopt corrupted COIL-100 and ALOI, while both original and corrupted images of CMU-PIE to testify these algorithms with the baseline, conventional AE [23]. The comparison results are shown in Table 4.1, where PIE-1 and PIE-2 denote the two views cases $\{C02, C14\}, \{C02, C27\}$ with its 10% corrupted versions PIE-1c and PIE-2c, respectively. ALOI-c represents the 10% corrupted data.

From the results, we could observe that LAE outperforms the conventional AE, that

Table 4.2: Recognition results (%) on CMU-PIE face database, where P1: {C02, C14}, P2: {C02, C27}, P3: {C14, C27}, P4: {C05, C07, C29}, P5: {C05, C14, C29, C34}, P6: {C02, C05, C14, C29, C31}. Red color denotes the best recognition rates. Blue color denotes the second best.

Original Images									
	PCA	LDA	RPCA+LDA	LatLRR	SRRS	LRCS	DAE	Ours-I	Ours-II
P1	69.03±0.08	70.46±0.05	74.39±0.08	77.92±0.03	78.27±0.04	87.78±0.02	85.65±0.12	87.97±0.06	88.04±0.08
P2	69.21±0.08	71.32±0.02	75.55±0.12	76.24±0.12	78.74±0.23	86.67±0.01	84.32±0.09	87.61±0.03	87.88±0.06
P3	68.52±0.12	63.51±0.75	75.29±0.09	75.29±0.07	77.45±0.02	87.38±0.19	84.53±0.04	87.87±0.09	88.01±0.06
P4	52.65±0.04	56.53±0.02	61.17±0.12	69.74±0.05	71.44±0.03	74.84±0.04	71.87±0.09	74.08±0.07	75.06±0.13
P5	34.94±0.08	24.07±0.25	38.66±0.08	42.54±0.12	38.86±0.02	44.48±0.03	42.32±0.07	44.42±0.10	45.35±0.09
P6	29.09±0.01	7.06±0.01	31.94±0.12	35.33±0.04	30.16±0.02	36.17±0.01	33.50±0.05	36.42±0.03	36.54±0.04
Corrupted Images with 10% Random Noise									
	PCA	LDA	RPCA+LDA	LatLRR	SRRS	LRCS	DAE	Ours-I	Ours-II
P1	64.87±0.32	26.71±0.20	73.07±0.11	73.10±0.07	72.27±0.05	78.98±0.03	77.14±0.11	81.02±0.08	81.54±0.07
P2	66.04±0.08	23.19±0.35	74.28±0.12	73.24±0.32	72.74±0.18	78.67±0.05	76.98±0.06	81.12±0.09	81.48±0.10
P3	65.21±0.04	20.34±0.75	73.92±0.12	73.85±0.12	71.45±0.08	78.38±0.26	77.32±0.09	81.94±0.12	82.31±0.08
P4	50.16±0.04	46.72±0.02	60.18±0.14	58.94±0.09	54.32±0.03	65.84±0.04	70.64±0.08	73.73±0.09	74.83±0.12
P5	31.74±0.08	6.67±0.25	37.65±0.09	39.26±0.12	32.34±0.02	39.48±0.03	40.32±0.09	43.92±0.08	43.81±0.09
P6	27.21±0.01	4.06±0.01	31.34±0.06	32.07±0.03	29.03±0.02	32.57±0.01	33.12±0.09	35.33±0.02	34.59±0.07

means jointly learning the low-rank dictionary could boost the deep feature learning of auto-encoder. Furthermore, we witness that our robust AEs with locality preserving low-rank dictionary could achieve better performance than LAE and AE for both unsupervised and supervised settings. That is, locality preserving property could generate more discriminative features for classification.

4.3.3 Comparison Experiments

We mainly compare with 1) traditional feature extract methods: PCA [64], LDA [65]; 2) low-rank based algorithms: RPCA+LDA [103], LatLRR [87], DLRD [107], LRCS [88], SRRS. Specifically, PCA, LDA, RPCA+LDA, LRCS and SRRS belong to dimensionality reduction algorithms so that we search the optimal dimensionality for each to report the performance. Besides, to further evaluate the effectiveness of our algorithm, DAE [30] is adopted as the baseline. For our algorithm, we have two modes, i.e., unsupervised mode (Ours-I), and supervised mode (Ours-II). Specifically, we set parameters $\alpha = 10^2, \lambda = 10^{-2}$. For DAE and our two modes, we apply a four-layer deep structure. For Ours-I, we set $k_1 = k_2 = 5$ for all cases, while for Ours-II, we set k_1, k_2 as the size of each class. We apply the nearest neighbor classifier (NNC) for all algorithms except DLRD and show experimental results in Table 4.2 and Figure 2.8(a).

From Table 4.2 and Figure 2.8(a), we could observe our proposed algorithm in two modes outperforms others in most cases, especially for the corruption cases. In the corruption cases,

our method has a significant improvement over others on two datasets (about 7% improvement on corrupted COIL dataset). All the algorithms suffer from additional noises; however, ours can still achieve appealing performance (only 1-2% performance degradation), which demonstrates the superiority of our method against noises in feature learning.

4.4 Conclusion

In this chapter, we developed a novel Deep Robust Encoder framework guided by a locality preserving low-rank dictionary learning scheme. Specifically, we designed a low-rank dictionary to constrain the output of the deep auto-encoder with corrupted input. In this way, the deep neural networks would generate more robust features by detecting noise from the corrupted data. Moreover, coefficient vectors z_i were maintained through the networks so that each output sample would be reconstructed by the most similar data samples in the dictionary with different weights. Furthermore, graph regularizers were developed to couple each layer's encoding to preserve more geometric structure. In experiments, we achieved more effective features for classification and results on several benchmarks demonstrated our method's superiority over other methods.

Chapter 5

One-Shot Face Recognition via Generative Learning

5.1 Background

One-shot face recognition is to recognize persons with only seeing them once. This problem exists in many real applications. For example, in the scenario of large-scale celebrity recognition, it naturally happens that some celebrities only have one or very limited number of images available. Another example is in the law enforcement scenario: it is usually the case that only one image of the personal ID is available for the target person.

The challenge of one-shot face recognition lies in two parts. First, a representation model is needed to transfer the face image into a discriminative feature domain. Although recent years have witnessed great progresses in deep learning for visual recognition, computer vision systems still lack the capability of learning visual concepts from just one or a very few examples [31]. A typical solution is to leverage many images from a different group of people (we call them *base set* and name the persons with limited number of training images *low-shot set*), and train a representation model using the images from the base set to extract face features for the images in the low-shot set.

Recently, there have been many research efforts focusing on training representation models with good generalization capability. Examples include [32, 33, 34, 35] etc., where face representation model is trained and tested across different groups of persons. However, improving the generalization and capability of face representation model is still an open problem which has attracted substantial effort in the area. When the distributions of the face dataset-A and face dataset-B are very different,

the representation model trained on dataset-A may not be discriminative enough on dataset-B. For example, if the data used to train a representation model do not include sufficient number of images for persons with a certain type of skin color, the trained model usually suffer from lower accuracy for those persons.

The second challenge of one-shot face recognition comes from estimating the partition for a given person in the feature space. A representation model transfers the face images of the same person into a cluster of dots in the feature space. To recognize all the faces for a given person, we need to estimate the shape, size, and location of the partition for this person in the feature space. However, with only one image (corresponding to one dot in the feature space), it is not easy to accurately and reliably estimate the distribution of the faces of the person to be recognized, which makes it challenging to estimate the boundary of the partition for this person in the feature space.

A straightforward yet a bit over-simplified way to estimate the partition for the one-shot person in the feature space is to assume each person claims a hyper-sphere with equal size in the feature space, and then use the k-nearest neighborhood (k-NN) classifier with a certain threshold to recognize persons each having only one image available. However, many recent works [110, 111, 112] demonstrate that directly using K-NN is often a sub-optimal solution compared with other methods which can learn the partition boundary in more informative way.

In order to better study the one-shot face recognition problem, there have been some benchmark tasks designed. One typical example is MegaFace in [113]. In this benchmarks task, one face image for a given celebrity is provided to search for another face image for this celebrity from up-to one million distractor images from regular persons. Since this task requires participants to extract feature vectors for all the face images and use 1-NN method to search for the image with the smallest distance, this task mainly focuses on evaluating the first challenge of the one-shot face recognition: to learn a discriminative face representation model.

Recently, there has been another benchmark task called MS-Celeb-1M: low-shot challenge [110] proposed. This task focuses on one-shot learning in the large-scale face recognition scenario. The MS-Celeb-1M low-shot challenge is to train a face recognizer to identify $21,000$ persons. For the $20,000$ persons among them (called *base set*, following the terminology defined in [114]) , about 50-100 training images per person are provided. For the other $1,000$ persons (denoted as *low-shot set*), only **one** training image per person is offered. The task is to study, with these training images only, how to develop an algorithm to recognize the persons in **both** data sets. The main focus of this task is the recognition accuracy for persons in the low-shot set as it shows the one-shot learning capability of a vision system, while also checking the recognition accuracy for those in the base set

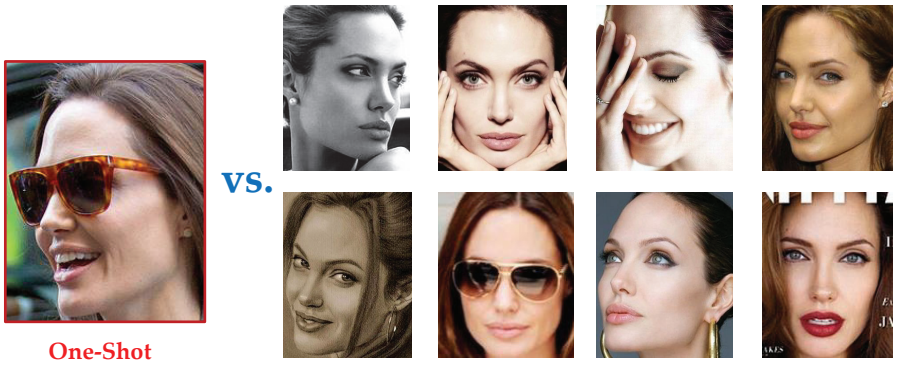


Figure 5.1: Illustration of One-Shot Challenge. The one-shot image in the leftmost column is used for training. The rest images (in the right panel) are the corresponding images for testing (partially selected from the test set). With only one image for each person, the challenge is how to recognize all these test images from hundreds of thousands of other testing images. More detailed results are presented in the experimental results section.

to ensure no harm to their performance.

MS-Celeb-1M: low-shot challenge evaluates large-scale low-shot face recognition system comprehensively. First, this benchmark task provides the base set to train a representation model, which is used to extract features not only for the base set but also for the low-shot set. This reveals the generalization capability of the representation model. Moreover, this benchmark focuses on the recognition accuracy for the low-shot set persons, which evaluates the quality of the estimation of the partition in the feature space for the low-shot persons. Last but not least, since the task is to recognize the persons in both the base set and the low-shot set, this task demands an algorithm which can handle serious data imbalance issues.

One-shot learning has attracted great attentions, attempting to make progress towards imparting this human ability to modern recognition systems [114, 115, 116, 110]. Generally, there are two ways to improve the low-shot face recognition performance. The first way is to enhance the generalization and discriminative capability of representation model. Examples include range loss [117], fisher face [118], center invariant loss [119], marginal loss [120], sphere face [121], etc. The second way is to improve the estimation of partitions in the feature space. There are two lines of strategies for this part. One is data augmentation, the other one is classifier adaptation. Along the first line, some human-designed data generation strategies are adopted to synthesize fake data for the low-shot/one-shot classes to boost the classification ability. Edwards et. al proposed handling the one-shot classification task by learning dataset statistics using the amortized inference of a variational

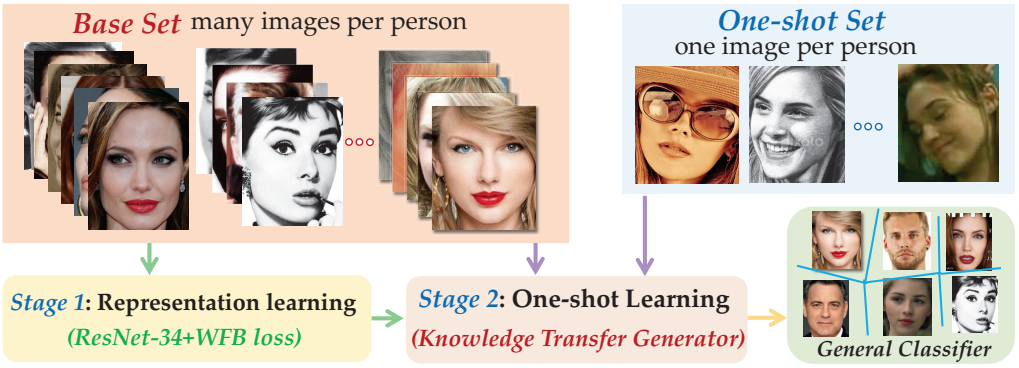


Figure 5.2: Illustration of one-shot face recognition problem with two phases. Stage 1: **Representation learning** phase seeks general face visual knowledge through training effective feature extractor using the base set. Stage 2: **One-shot GAN learning** phase builds a general classifier to recognize persons in both the base set and the one-shot set based on the deep features.

auto-encoder [122]. Hariharan et al. designed a method for hallucinating additional examples for the data-starved novel classes [114]. Mehrotra et al. presented an additional generator network based on the Generative Adversarial Networks where the discriminator is the proposed residual pairwise network [116]. For the second line, the idea is to adapt the base classifier to the novel classifier through multi-layer transformations or some specific loss to boost the classifier space of one-shot classes. Guo et al. proposed a novel supervision loss named as Underrepresented-classes Promotion (UP) loss term, which aligns the norms of the weight vectors of the one-shot classes (a.k.a. underrepresented-classes) to those of the normal classes [110], which directly boosts the one-shot classifier parameters without considering the difference of class variance for different one-shot classes. However, human-designed generation rules cannot learn the data distribution well to synthesize data to effectively improve the recognition. Therefore, the existing one-shot research efforts fail to jointly seek a general classifier when automatically generating augmented data.

Due to the challenges and fast developments of one-shot face recognition in real-world scenarios, it is essential to provide a comprehensive review of this area for novices. To this end, we first systematically review the progress of one-shot face recognition in the wild, included with detailed descriptions of our recent work, large-scale face recognition system (Figure 5.2) with extended experimental results. This system achieves appealing performance for one-shot classes while keeping base classes at a very high Top1 accuracy. We target at seeking a general classifier for both the base and novel classes. The core idea of our model is to generate effective auxiliary data for the one-shot classes, and thus we can span the feature space for the one-shot classes to facilitate

the one-shot face recognition. To our best knowledge, this is one of the first works to explore the generative model in a general classifier learning for one-shot challenge. Specifically, we focus on the one-shot learning stage, where we attempt to train a more powerful classifier for both base and novel classes. Finally, we also provide several interesting and challenging future directions for one-shot face recognition. The main contributions of our paper are listed in three folds as follows:

- We systematically review the progress of one-shot face recognition to provide readers an efficient way to understand the one-shot challenges and state-of-the-art one-shot face recognition algorithms. We further discuss the relationship between one-shot learning and zero-shot learning as well as transfer learning.
- We jointly incorporate generative adversarial networks in training a general classifier for both base and novel classes. In detail, the generator attempts to synthesize more effective fake data for the one-shot classes to enrich the data space of one-shot classes, while the discriminator is built to guide the face data generation to mimic the data variation of base classes and adapt to generate novel classes.
- We design generative adversarial networks in the feature domain which we first obtain by training a deep ConvNet model on the base classes [1]. More specifically, we build the conditional generative adversarial networks with an auxiliary classifier to augment more effective features and enhance the general classifier learning for one-shot classes.
- We evaluate our proposed model on a large-scale one-shot face dataset, and achieve significant improvement in the one-shot classification with coverage rate 94.98% at the precision of 99%. Meanwhile, our model can still achieve very appealing performance as Top1 accuracy of 99.80% for the base classes.

5.2 One-Shot Face Recognition: A Review

In this section, we mainly review the current status of one-shot learning in the literature. Before that, we first provide the challenges for one-shot learning. Moreover, we also discuss two related topics to one-shot learning, i.e., zero-shot learning and transfer learning.

[1] We train generative model on feature domain instead of image domain for the following reasons. Typically, image synthesis is a more challenging task than image classification/recognition. Especially for the current generative models, it is still an open problem to generate meaningful faces with high quality in many cases [123, 124]. This might be the reason that we don't find any existing one-shot learning work using generative models to synthesize face images. Furthermore, we joint our generative model and the deep architecture into a unified model to seek more general feature extractor.b

5.2.1 One-Shot Challenges

With remarkable success of deep learning in computer vision applications [125, 99], the frontier of face recognition research has also been significantly advanced [121, 35, 34, 126, 32, 127, 128]. Generally, face recognition can be formulated into two stages. The first stage is face feature extraction, and the second step is to estimate the person's identity from the extracted face feature. Nowadays, we notice the major focus point in face recognition is to seek a discriminative face feature extractor. In this scenario, a face feature extractor is typically trained on face images from a group of persons, while evaluated on images from a different group of persons in the verification or identification task. For example, the verification task with the LFW dataset[2] is the de facto standard test to evaluate the generalizablity of face features, though the performance on this dataset is getting saturated. Moreover, a lot of face identification tasks, e.g., MegaFace[3] or LFW with the identification setup, are essentially to evaluate face features since the identification is achieved by comparing face features between query and gallery images.

The major merit of the above setup is that the generalization capability of face representation model can be clearly evaluated, since the persons in the training phase are different from the persons in the test phase. This is very important when the images of the target persons are not accessible during the training phase. Unfortunately, we observe the best performance for the above setup is typically obtained by using very large, private dataset(s), which makes it impossible to reproduce these work.

Moreover, though to obtain a good feature extractor is essential and critical for face identification, good feature extractor is not yet the final solution for the identification. Despite its success,

learning There are only few previous works that discuss about learning from data imbalance in the context of deep face recognition. When there are many persons to be recognized, it naturally happens that for some of the persons to be recognized, there might be very limited number of training samples, or even only one sample for each of them. This is very useful when the images for the target persons are available beforehand, because it generally leads to better performance to train with images for the target persons compared with to train with images for other persons (assuming similar total amount of images). When we can have access to one training sample for the target persons, it becomes one-shot face recognition.

[2]http://vis-www.cs.umass.edu/lfw/
[3]http://megaface.cs.washington.edu/

One of the major challenge for one-shot learning is caused by the highly imbalanced training data. The low coverage for the one-shot classes is related to the fact that the only one sample in each one-shot class occupies a much smaller partition in the feature space, compared with the samples in each base class (see Figure 5.1). This is because a class with one sample usually has a much smaller (even 0 for one sample) intra class variance than a classes with many samples which can span a larger area in the feature space. Besides this unique challenge, there are also other challenges introduced by the fact that different persons may have very similar faces, and the fact that the faces from the same person may look very different due to lightning, pose, and age variations.

5.2.2 One-Shot Face Recognition Revisit

In the general image recognition domain, the recent low-shot learning work [114] also attracts a lot of attentions. Their benchmark task is very similar to one-shot face recognition but in the general image recognition domain: the authors split the ImageNet data[4] into the base and low-shot (called novel in [114]) classes, and the target is to recognize images from both the base and low-shot classes. Their solution is quite different from ours since the domain is quite different. We will not review their solution here due to the space constraint, but list results from their solution as one of the comparisons in the experiments section.

Overall, one-shot learning is still an open problem. A natural source of information comes from additional data via "data manufacturing" [129] in various ways. In a broad sense, learning novel classes is addressed by exploiting and transferring knowledge gained from familiar classes. This is to imitate the human ability of adapting previously acquired experience when recognizing novel classes. In the following, we will revisit different categories of one-shot learning methods, including the most popular "general feature learning", classifier learning & adaption, and data augmentation.

The objective of one-shot face recognition is to measure the recognition ability of a model across classes with one training sample. Specifically, the model is trained on labeled training data with two sets without identity overlap, i.e., **Base Set** (i.e., normal classes) $\{X_b, Y_b\}$ with c_b classes and **Novel Set** (i.e., one-shot classes) $\{X_n, Y_n\}$ with c_n classes. The goal is to build a general c-class recognizer ($c = c_b + c_n$). In this paper, we will mainly focus on the performance on the novel classes while keeping an eye on the accuracy for base classes.

[4] http://www.image-net.org/

5.2.2.1 Generalized Feature learning

Cross entropy with Softmax has demonstrated good performance in supervising the face feature extraction model training. In order to further improve the performance of representation learning, many methods have been proposed to add extra loss terms or slightly modify the cross entropy loss (used together with Softmax for multinomial logistic regression learning) to regularize the representation learning in order to improve the feature discrimination and generalization capability.

Among all these works, we consider the center loss [128] as one of the most representative one (a similar idea published in [130] during the same time). In [128], face features from the same class are encouraged to be close to their corresponding class center (actually, approximation of the class center, usually dynamically updated). By adding this loss term to the standard Softmax, the authors obtain a better face feature representation model [128].

There are many other alternative methods, including the range loss in [117], fisher face in [118], marginal loss in [120], sphere face in [121], etc. Each of these methods has its own uniqueness and advantages under certain setup. Guo et al. designed a different kind of loss term adding to the cross entropy loss of the Softmax to improve the feature extraction performance [110]. Compared with center loss in [128] or sphere face in [121] (these two are the most similar ones), Guo et al. demonstrated that their proposed method has better performance from the perspective of theoretical discussion and experimental verification. Unfortunately, it is not very practical to compare all these cost function design methods fairly and thoroughly, since these cost functions were implemented with different networks structures, and trained on different datasets. Sometimes parameter adjustment is critically required when the training data is switched. We will work on evaluating more methods in the future.

5.2.2.2 k-Nearest Neighbor & Softmax Classifier

After a good face feature extractor is obtained, the template-based method, e.g., k-nearest neighborhood (k-NN) classifier, is widely used for face identification these days. The advantages of k-NN is clear: no classifier training is needed, and k-NN does not suffer much from imbalanced data, etc. However, experiments in [112, 131, 110, 111] demonstrate that the accuracy of k-NN with the large-scale face identification setup is usually lower than Softmax Classifier, when the same feature extractor is used. Moreover, if we use all the face images for every person in the gallery, the complexity is usually too high for large scale recognition, and the gallery dataset needs to be

very clean to ensure high precision. If we do not keep all the images per person, how to construct representer for each class is still an open problem.

As described above, Softmax classifier demonstrates overall higher accuracy compared with k-NN in many previous publications. This is mainly because in Softmax classifier, the weight vectors for each of the classes is estimated using discriminant information from all the classes, while in the k-NN setup, the query image only needs to be close enough to one local class to be recognized. Moreover, after feature extraction, with Softmax classifier, the computational complexity of estimating the persons' identity is linear to the number of persons, not the number of images in the gallery. However, the standard Softmax classifier suffers from the imbalanced training data and has poor performance with the low-shot classes even these classes are oversampled, though the overall accuracy is higher than k-NN. Recently, some works develop hybrid solutions by combining Softmax classifier and k-NN [131, 111] and achieve promising results. In these work, when Softmax classifier does not have high confidence (threshold tuning is needed), k-NN is used.

We solve this problem from a different perspective. Different from the hybrid solution, our solution only has one Softmax classifier as the classifier so that no threshold is needed to switch between classifiers. We boost the performance of Softmax classifier by involving the generated data.

5.2.2.3 Classifier Adaptation

Another type of knowledge transfer focuses on modeling (hyper-)parameters that are shared across domains, typically in the context of generative statistical modeling [132, 133, 134]. Li et al. operated in a variational Bayesian framework by incorporating previously learned classes into the prior and combining with the likelihood to yield a new class posterior distribution [132, 135]. Gaussian processes and hierarchical Bayesian models are also employed to allow transferring in a non-parametric Bayesian way. Specifically, hierarchical Bayesian program learning utilizes the principles of compositionality and causality to build a probabilistic generative model of visual objects [136, 31]. In addition, adaptive SVM and its variants present SVM-based model adaptation by combining classifiers learned on related categories [115, 137]. Wang et al. assumed there exists a generic, category agnostic transformation from small-sample models to the underlying large-sample models [138], and thus they explored a novel learning to learn approach that leverages the knowledge gained when learning models in large sample sets to facilitate recognizing novel categories from few samples. Despite many notable successes, it is still unclear what kind of underlying structures are shared across a wide variety of categories and are useful for transfer.

5.2.2.4 Data Augmentation

Data augmentation is a straightforward way to boost the one-shot class performance, since it could compensate the shortage of one-shot class samples by synthesizing more data. However, it is the key to generate meaningful data with enough data variance for one-shot classes. Along this line, Hariharan et al. presented a way of "hallucinating" additional examples for one-shot (low-shot) classes by transferring modes of variation from the base classes [114]. Their experiments demonstrated those additional examples improve the one-shot top-5 accuracy on low-shot classes while also maintaining accuracy on the base classes. Note that Hariharan et al. adopted some human-designed rules to augment the data space for low-shot classes, which is very complexed and not easily spread in real-world applications [114].

Most recently, generative models [139, 123] are exploited to synthesize more training data for one-shot classes by automatically capturing the data variance from base classes. Specifically, Rezende et al. developed a class of sequential generative models by combining the representational power of deep learning with the inferential power of Bayesian reasoning, which is among the state-of-the art in density estimation and image generation [140]. Choe et al. adapted a generator to increase the size of training dataset, which includes a base set, a widely available dataset, and a novel set, a given limited dataset, while adopting transfer learning as a backend [141]. Mehrotra et al. proposed a deep residual network with an additional generator network that allows the efficient computation of this more expressive pairwise similarity objective [116]. Our proposed generative model also belongs to this category. The key difference from previous work is we exploit data generation in feature domain instead of image domain by jointly seeking a general classifier. Moreover, we involve the class center and class variance to synthesize more efficient data. Moreover, this work is our previous conference extension [142]. In this journal extension, we systematically review the recent progress on one-shot learning, also we compare with zero-shot learning and transfer learning, further we propose some new directions for one-shot face recognition. Besides, we improve our representation learning model with WFB loss and generative model with knowledge transfer on data variance.

5.2.3 Beyond One-Shot Learning

In this section, we will briefly discuss two lines of relate works to one-shot learning, which are zero-shot learning and transfer learning. The main shared point across three tasks is to transfer knowledge across different sets. However, they follow in different settings.

5.2.3.1 Zero-shot Learning

Zero-shot learning (ZSL) manages to build models of visual concepts without test data of the concepts. Since visual information from such test classes is unavailable in the training stage, ZSL requires auxiliary information to compensate for the unobserved visual information. Attribute-based descriptions are the most commonly-used characteristics shared across various classes [143, 144, 145, 146, 147], which provide an intermediate representation to connect the low-level visual features with the semantic labels. Given the low-level visual representations of images and their underlying high-level semantics, the core issue in ZSL turns to "how to adapt knowledge from the visual data of seen classes to those of unobserved ones" [148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158].

The general setting of ZSL is described as follows: Suppose there are C seen classes with n labeled samples $\mathcal{S} = \{X, A, y\}$ and C_u unseen classes with n_u unlabeled samples $\mathcal{U} = \{X_u, A_u, y_u\}$. Each sample is denoted as visual feature with dimension d. Assume there are n samples in the seen training data and n_u samples in the unseen test data, and thus, the visual features are represented as $X \in \mathbb{R}^{d \times n}$ and $X_u \in \mathbb{R}^{d \times n_u}$, while their corresponding class label vectors are $y \in \mathbb{R}^n$ and $y_u \in \mathbb{R}^{n_u}$. In ZSL setting, the observed and unobserved classes have no label overlap, i.e., $y \cap y_u = \emptyset$. $A \in \mathbb{R}^{m \times n}$ and $A_u \in \mathbb{R}^{m \times n_u}$ are the m-dimensional semantic representations of instances in the seen and unseen datasets, respectively. For the seen dataset, A is provided in advance since seen samples X are labeled with either attribute features or word2vector representations corresponding to their class labels y. On the other hand, A_u needs to be estimated since the unseen data are unlabeled. The task of ZSL is to predict A_u and y_u given visual features X_u using the classifier learned from seen classes.

5.2.3.2 Transfer Learning

Transfer learning (TL) has been witnessed as an appealing technique in many real-world applications in computer vision and pattern recognition. Specifically, transfer learning technique is designed to address the problem when the distribution of the source domain is different from that of the target domain [11]. Thus, key problems turn to be adapting either source or target domain, or both of them to mitigate the distribution differences of two domains [12, 159, 13, 14, 160]. Generally, domain can be defined as probability distribution $\mathbb{P}_{XY}$ on $\mathcal{X} \times \mathcal{Y}$, in which $\mathcal{X}$ and $\mathcal{Y}$ represent the data and label spaces, respectively. For simplicity, we denote $\mathbb{P}_{XY}$ as $\mathbb{P}$. Assume $\mathbb{D} = \{x_j; y_j\}_{j=1}^n$ as an independent and identically distributed (i.i.d.) sample from a domain. In transfer learning, there are several sub-topics, e.g., self-taught learning, domain adaptation, domain generalization.

First of all, we assume $\Delta = \{\mathbb{P}_{s,1}; \cdots ; \mathbb{P}_{s,m}\}$ as a combination of m source domains and $\mathbb{P}_t \notin \Delta$ as a target domain. Assume i-th source domain $\mathbb{D}_{s,i} = \{x_i^j; y_i^j\}_{j=1}^{n_i}$ with n_i labeled samples. For domain adaption, we generally have a single source, i.e., $m = 1$, and we also have target domain $\mathbb{D}_t$ with the same categories in the training stage. In domain generalization [75], we only have m source domains in the training stage. In the test stage, we aim to evaluate on some unseen domains $\mathbb{D}_t$ with the same categories. The main difference between multi-source domain adaptation [20] and domain generalization is on the accessibility of the target data during the training stage. Both manage to seek a labeling function $\theta : \mathcal{X} \to \mathcal{Y}$ that can achieve good performance on the target data. Note that domain generalization would definitely become to multi-source domain adaptation, when $\mathbb{P}_t \in \Delta$. Although these two are highly related problems, domain adaptation approaches generally cannot be directly exploited in domain generalization scenario. Thus, it is significantly desirable to propose models efficiently for domain generalization.

5.2.3.3 Comparison with ZSL and TL

In this section, we would discuss the key difference between one-shot learning to zero-shot learning and transfer learning.

Zero-shot learning is different from one-shot learning in the following aspects. First, zero-shot learning assumes that the test data are totally unavailable in the training stage. While one-shot learning considers there is only one sample available for those classes in the training stage. Second, there is an intermediate domain in zero-shot learning, i.e., semantic features (attributes), to link the training (seen) and test (unseen) data, which can be considered as a two-view learning task. While one-shot learning only exists the visual data. Finally, in zero-shot learning, the visual data for training and test classes are sample from different distributions, while the visual data in one-shot learning are drawn from the similar or same distribution.

Generally, in transfer learning, there are two domains with different distributions, i.e., source domain and target domain. The goal is to transfer the knowledge from source domain to boost the target domain learning. However, in one-shot learning, the goal is to adapt the knowledge from base classes to one-shot classes. In this sense, we consider base classes as source domain while one-shot classes as target domain. But, the major difference is base classes and one-shot classes are sampled from the same or similar distribution. On one hand, the challenge for one-shot learning is that one-shot classes are under insufficient sampling such that they are difficult to be recognized. On the other hand, the challenge for transfer learning is that distribution divergence across source and

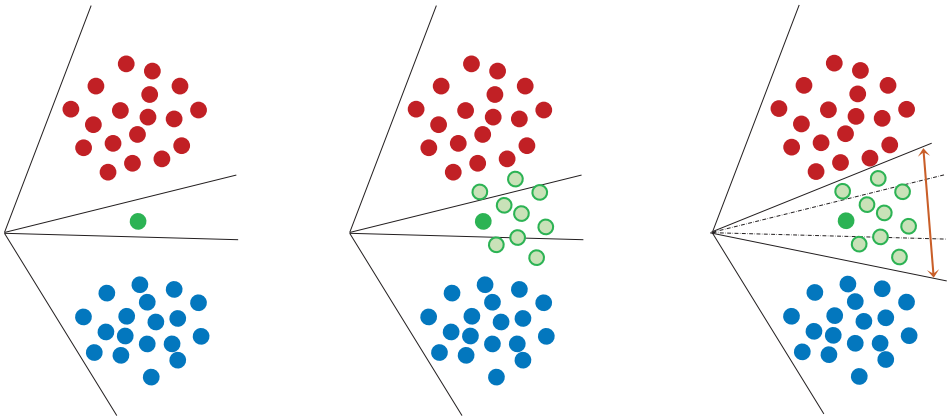


Figure 5.3: Illustration of generative model to synthesize more samples for one-shot classes then update decision boundary bias. (a) in the beginning, we have one training sample for one-shot class while many samples for base classes, thus the classifier would be dominated by the bias. (b) we explore our generate model to synthesize more samples for one-shot class. (c) with the augmentation of feature space for one-shot classes, the classifier also be updated with its one-shot classifier space enlarged.

target domains so that we need to mitigate the domain mismatch for effective knowledge transfer.

5.3 The Proposed Algorithm

In this section, we will first introduce the motivation of our proposed model for one-shot face recognition, then present the framework details as well as the training process.

Specifically, our solution to one-shot face recognition includes the following two phases (Figure 5.2). The first phase is named as *representation learning*. In this phase, we build face representation model using all the training images from the *base set*. The second phase is called as *one-shot learning*. In this phase, we train a multi-class classifier to recognize the persons in both *base set* and *one-shot set* based on the representation model learned in phase one. We design a generative one-shot learning model to improve the recognition performance for the persons in the one-shot set.

5.3.1 Motivation

One-shot face recognition is challenging due to limited samples during model training, while general deep frameworks treat base and one-shot class equally, which leads to biased updates

of the recognition model. Thus, it is essential to generate more effective data to improve the ability of the general classifier. Traditional data augmentation strategies [114] only adopt human designed rules to generate more data for the one-shot classes. Hence, the enhancement to the classifier is limited. Another challenge is that it usually hurts the base classification when we try to improve the classification ability for one-shot classes. That is, the learned classifier is impractical in real-world applications when dealing with a general face recognition problem. Hence, it is essential to balance these two sets.

Moreover, generative models are very popular due to its promising ability to synthesize effective data automatically, which are similar to the real data with a guidance from a discriminator. While for one-shot face recognition, it is essential to generate effective data with large variations for the one-shot classes in order to span their classifier space. Generally, data from the base classes have large within-class variations, and thus, it is helpful to adapt the variations of base classes to the one-shot classes during data generation. To generate more meaningful data for one-shot classes, we jointly seek a general classifier with the input of real data and fake data. Such a joint learning framework could benefit generating meaningful data and improving the classification ability, specially updating the decision boundary of the classifier shown in Figure 5.3, where more meaningful data are augmented to enlarge the feature space then updating the classifier boundary.

5.3.2 Representation Learning

To learn more effective feature representation, we design our face representation model with supervised learning framework considering persons' identities as class labels. Specifically, we propose the loss function as follows:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_a \,, \tag{5.1}$$

where $\mathcal{L}_s$ is the standard cross-entropy loss used for the Softmax layer, while $\mathcal{L}_a$ is the newly proposed loss used to improve the feature discrimination and generalization capability. λ is the trade-off parameter between two loss functions.

More specifically, we recap the first term, cross-entropy $\mathcal{L}_s$ as follows:

$$\mathcal{L}_s = -\sum_i \sum_k t_{k,i} \log p_k(x_i), \tag{5.2}$$

where $t_{k,i} \in \{0, 1\}$ is the ground truth label indicating whether the i-th image belongs to the k-th class, and the term $p_k(x_i)$ is the estimated probability that the image x_i belongs to the k-th class,

defined as,

$$p_k(x_n) = \frac{\exp\left(\mathbf{w}_k^\top \boldsymbol{\phi}(x_i)\right)}{\sum_i \exp\left(\mathbf{w}_k^\top \boldsymbol{\phi}(x_i)\right)}\,, \tag{5.3}$$

where $\mathbf{w}_k$ is the weight vector for the k-th class, and $\boldsymbol{\phi}(\cdot)$ denotes the feature extractor for image x_n. Note that in all of our experiments, we always set the bias term $b_k = 0$. We choose the standard residual network with 34 layers (ResNet-34) [125] as our feature extractor $\boldsymbol{\phi}(\cdot)$ using the last pooling layer as the face representation. ResNet-34 is used due to its good trade-off between prediction accuracy and model complexity, yet our method is general enough to be extended to deeper network structures for even better performance. We have conducted comprehensive experiments and found that removing the bias term from the standard Softmax layer in deep convolutional neural network does not affect the performance. However, it is worth noting that this leads to a much better understanding of the geometry property of the classification space.

The second term $\mathcal{L}_a$ in the cost function (Eq. (5.1)) is defined as

$$\mathbf{w}'_k \leftarrow \mathbf{w}_k \tag{5.4}$$

$$\mathcal{L}_a = -\sum_k \sum_{i \in C_k} \frac{\mathbf{w}_k^{'\top} \boldsymbol{\phi}(x_i)}{\|\mathbf{w}'\|_2 \|\boldsymbol{\phi}(x_i)\|_2}\,. \tag{5.5}$$

We set the parameter vector $\mathbf{w}'_k$ to be equal to the weight vector $\mathbf{w}_k$. This loss term encourages the face features belong to the same class to have similar direction as their associated classification weight vector $\mathbf{w}_k^\top$. We name this loss term as Weights-guided Feature vector Bundling (WFB). Calculating the derivative with respect to $\boldsymbol{\phi}(x_i)$, we have

$$\frac{\partial \mathcal{L}_a}{\partial \boldsymbol{\phi}(x_i)} = \frac{1}{\|\boldsymbol{\phi}(x_i)\|_2} \left(\frac{\mathbf{w}_k^{'\top}}{\|\mathbf{w}'_k\|_2} - \frac{\boldsymbol{\phi}(x_i)^\top \cos\theta_{i,k}}{\|\boldsymbol{\phi}(x_i)\|_2} \right)\,, \tag{5.6}$$

where $\theta_{i,k}$ is the angle between $\mathbf{w}'_k$ and $\boldsymbol{\phi}(x_i)$. Note that $\mathbf{w}'_k$ in this term is the parameter copied from $\mathbf{w}_k$, so there is no derivative to $\mathbf{w}'_k$. For experiment ablation purpose, we also tried to back propagate the derivative of $\mathbf{w}_k$, but did not observe better results.

Discussion: There have been a lot of effort in integrating extra terms with cross-entropy loss to improve the feature generalization ability. The most similar version is the center loss in [128], also known as the dense loss in [130] published during the same time. In center loss, the extra term is defined as

$$\mathcal{L}_c = -\sum_k \sum_{i \in C_k} \|\mathbf{c}_k - \boldsymbol{\phi}(x_i)\|_2^2\,, \tag{5.7}$$

where $\mathbf{c}_k$ is defined as the *class* center (might be dynamically updated as the approximation of the true class center due to implementation cost).

Our method is different from center loss from two perspectives. First, minimizing the cost function (Eq. (5.7)) may lead to two consequences. While it helps reduce the discrepancy between $\phi(x_i)$ and its associated center $\mathbf{c}_k$, it also reduces the norms of $\phi(x_i)$ and $\mathbf{c}_k$. The second consequence is usually not good as it may hurt the classification performance. We did observe in our experiment that over training with center loss would lead to features with too small norms and worse performance compared with not using center loss (also reported in [128]). On the contrary, our loss term only considers the angular between $\phi(x_i)$ and $\mathbf{w}'_k$, and will not affect the norm of the feature. In our experiment section, we demonstrate that our method is not sensitive to the parameter tuning.

Second, please note that we use the weight vector in Softmax $\mathbf{w}_k$ to represent the **classification** center, while in (5.7), the variable $\mathbf{c}_k$ is the **class** center. The major difference is that $\mathbf{w}_k$ is updated (naturally happens during minimizing $\partial\mathcal{L}_c$) using not only the information from the k-th class, but also the information from the other classes. In contrast, $\mathbf{c}_k$ is updated only using the information from the k-th class (calculated separately). More specifically, according to the derivative of the cross-entropy loss in Eq. (5.2),

$$\frac{\partial\mathcal{L}_s}{\partial\mathbf{w}_k} = \sum_i (p_k(x_i) - t_{k,i})\phi(x_i)\,, \tag{5.8}$$

the direction of $\mathbf{w}_k$ is close to the direction of the face features in the k-th class, and being pushed far away from the directions of the face features *not* in the k-th class.

5.3.3 Generative One-Shot Learning

Generative adversarial networks (GANs) [139] consist of two neural networks trained in opposition to each other. The basic GANs framework can be augmented through side information. One strategy is to supply both the generator and discriminator with class labels or latent information to obtain class conditional samples [123]. Class conditional synthesis can significantly improve the quality of generated samples [161]. Richer external information such as image captions and bounding box localizations may improve sample quality further [162]. Instead of feeding auxiliary information to the discriminator, one can train the discriminator with reconstructing side information. This is done by modifying the discriminator to contain an auxiliary decoder network that outputs the class label for the training data [163, 164] or a subset of the latent variables from which the samples are synthesized [124]. It is well-known to improve performance on the original task by

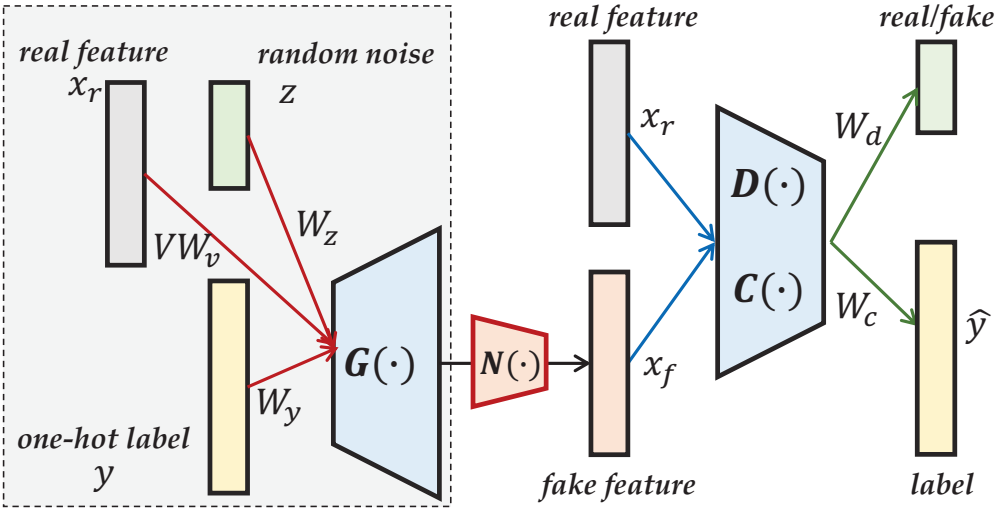


Figure 5.4: Illustration of generative one-shot face recognizer, where z is the random noise vector, y is the one-hot label, $x_r = \phi(x)$ is the real feature, while x_f is the generated fake feature. $G(\cdot)$ is the generator with the input of random noise z, original real feature x_r, and one-hot label y. The output of generator with normalization $N(\cdot)$ will achieve the fake feature x_f. $D(\cdot)$ is the discriminator which aims to differentiate the real and fake features, while $C(\cdot)$ is the a general multi-class classifier.

forcing a model to perform additional tasks [165]. Additionally, an auxiliary decoder could leverage pre-trained discriminators (e.g., image classifiers) for further enhancing the synthesized images [166]. Conditional generative models prove to be more effective to synthesize meaningful data [123].

Given random noise $z \in \mathbb{R}^{d_z}$, real feature $\phi(x) \in \mathbb{R}^{d_x}$ with its one-hot label $y \in \mathbb{R}^{d_y}$. In the generator, the prior input noise $p_z(z)$, and one-hot label y are combined in joint hidden representation, and the adversarial training framework allows for considerable flexibility in how this hidden representation is composed. In the discriminator, $\phi(x)$ and y are presented as inputs and to a discriminative function, where $\phi(\cdot)$ denotes the feature extractor for sample x. The objective function of a two-player minimax game is as follows:

$$\mathcal{L}_d^f = \mathbb{E}[\log(1 - D(G(z|y)))]$$
$$\mathcal{L}_d^r = \mathbb{E}[\log(D(\phi(x)))] \tag{5.9}$$

where the generator aims to make the generated features similar to real features, attempting to minimize $\mathcal{L}_d^f$; the discriminator aims to differentiate the real and fake features by maximizing $\mathcal{L}_d^r + \mathcal{L}_d^f$.

5.3.3.1 Knowledge Transferable Generator

As we know, the above conditional generative model does not involve the variance of base classes, which may not effectively adapt the knowledge from base classes to one-shot classes. We assume that base and one-shot classes share the same true distribution. While instances in the base class well-sample the true distribution, the one-shot instances are under sampling. By transferring the intra-class variance of a base class to a one-shot class, the feature distribution within the one-shot class can be enriched to be similar to base classes. The task then is to model the intra-class variance. By assuming the variance to be a multi-variate Gaussian, Cao et al. proposed a joint Bayesian model that takes the facial feature as the identity feature plus the intra-class variance [167]. From the observation of researchers, the identity feature is well-approximated by the feature center. Thus, the facial feature is represented as:

$$\boldsymbol{\phi}(x_i) = \mathbf{c}_i + v_i \tag{5.10}$$

where c_i is i-th class feature center, $\boldsymbol{\phi}(x_i)$ is one arbitrary sample in the class while v_i represents the variance from arbitrary feature to the class center.

Different from Cao's assumption for the variance [167], we completely rely on the data variance from the base classes. That follows the human cognitive process that we could adapt the previous knowledge (data variance from base classes) to learn new things (one-shot classes). Here we are not to model the variance across all the classes but rather to model the one-shot class variance. Given the observed feature distribution in base classes, we seek a parameterization to transfer the variance to the one-shot classes. Theoretically, any complete space decomposition method is valid here. In this paper we incorporate such knowledge transfer into the generator $G(\cdot)$ to simulate for effective features for one-shot classes.

Specifically, we design the generator $G(\cdot)$ with the input of random noise $z \in \mathbb{R}^{d_z}$, intra-classes variance of base classes $V \in \mathbb{R}^{d \times d}$ (later we will discuss how to obtain this matrix), and one-hot label $y \in \mathbb{R}^c$.

$$
\begin{aligned}
G(z|V,y) &= f_1(W_g \begin{bmatrix} z \\ y \end{bmatrix} + Vv_i) \\
&= f_1([W_z, W_y] \begin{bmatrix} z \\ y \end{bmatrix} + Vv_i) \\
&= f_1(W_z z + W_y y + Vv_i),
\end{aligned} \tag{5.11}
$$

where $W_g = [W_z, W_y]$ while $W_z \in \mathbb{R}^{d \times d_z}$ and $W_y \in \mathbb{R}^{d \times c}$. $v_i \in \mathbb{R}^d$ is a coefficient vector to find the most related intra-class variances from the base class (V). To go deeper, we consider the variance of ons-shot classes could be represented by its nearby base classes. And $f_1(\cdot)$ is the element-wise activation function, e.g., ReLU function or Sigmoid function.

Remark: For the generator $G(\cdot)$, we attempt to synthesize meaningful data to augment the one-shot classes. The goal is to span the feature space of one-shot classes around its center features. Generally, we can calculate the mean feature of base classes as their centers, while the one-shot features for novel classes are usually not the centers and may be far away from their centers. From Eq. (5.11), we notice there are three parts used to generate the fake features. There two parts $W_z z + W_y y$ are from the conventional conditional generator [123], which aims to seek two projections, one is for random noise and the other for conditional one-hot label. The third part Vv_i follows a dictionary-based reconstruction format, that is, we hope to select the most relevant data variances from the base classes to synthesize one-shot feature. That is, we consider there are d bases for class variance which can be well captured from base classes. Thus, the class variance for one-shot classes can be represented the combination of the bases in specific coefficients. On one hand, this term can adapt the knowledge of class variance from base classes to one-shot classes to synthesize meaningful data for one-shot classes. On the other hand, this term is able to better estimate the one-shot class centers since we only have one training sample in advanced so that we cannot obtain good one-shot class centers in the beginning. For simplicity, we adopt Principal Component Analysis (PCA) to parametrize the intra-class variance of regular classes, i.e., we first build a matrix $\mathcal{V} = [\boldsymbol{\phi}(x_j^i) - c_i]_{j,i} \in \mathbb{R}^{d \times n_b}$ ($d \ll n_b$), then we select d eigen-vectors with large eigen-values to represent the dictionary V.

However, our reconstruction term Vv_i makes the sample dependent when optimizing v_i. Thus, we explore a projective dictionary to approximate v_i with $W_v \boldsymbol{\phi}(x_i)$. In this way, we only need to optimize a shared projective dictionary $W_v \in \mathbb{R}^{d \times d}$ for all classes. Hence, the generator can be

reformulated as:

$$G(z|V, \boldsymbol{\phi}(x), y) = f_1\Big(W_z z + W_y y + V W_v \boldsymbol{\phi}(x)\Big), \tag{5.12}$$

Moreover, we hope $W_y y$ could keep the class center information, while $W_z z$ to compensate the residual information. Therefore, we initialize W_y with the class-center features, and then we would get its class center by multiplying W_y and its one-hot label y. Specifically, the base part is initialized with the mean features of c_b classes, while novel part is initialized with the available one-shot features. *Note that the one-shot class centers are not the real centers, so that we hope the intra-class variance part could facilitate to optimize the W_y, aseptically the novel part.* For random noise part, we initialize W_z and z randomly. In this way, we can capture the data variation within base classes and adapt to generate more meaningful data for novel classes. Another thing is that the scale of $W_y y$ is the same as that of x, and thus, the scale would be improved if we add a random part $W_z z$. Therefore, we add a normalization process to make the fake feature with the same scale to the real feature. Specifically, we define our normalization process as

$$N(G(z|V, \boldsymbol{\phi}(x), y)) = \frac{\alpha G(z|V, \boldsymbol{\phi}(x), y)}{\|G(z|V, \boldsymbol{\phi}(x), y)\|_2}, \tag{5.13}$$

where α is the mean norm of real feature across all samples.

5.3.3.2 Joint Discriminator & Classifier

The generator $G(\cdot)$ attempts to augment more training data for one-shot classes, which is guided by the discriminator $D(\cdot)$. The goal of discriminator manages to differentiate the fake features and real features. In this way, two players could compete with each other to synthesize more effective data. Since we build the generator in the feature domain, we design one-layer fully-connected network to build the discriminator as follows:

$$D(x) = f_2(W_d \bar{x}), \tag{5.14}$$

where $W_d \in \mathbb{R}^{1\times d}$ and $f_2(\cdot)$ projects $\bar{x}$ to a scale between 0 and 1. $\bar{x}$ can be the real feature or generated feature. For real feature, the output of $D(\bar{x})$ tends to be 1, otherwise 0.

Simultaneously, we target at building a general classifier $C(\cdot)$ across c classes for both base classes and one-shot classes based on the real feature and synthesized feature. Specifically, we adopt the standard Softmax classifier with loss function $\mathcal{L}_s$ defined as (Eq. (5.2)).

To sum up, we propose our generative one-shot learning model by incorporating generator, discriminator and classifier together into a unified framework. Specifically, there are two players,

i.e., $D(\cdot) + C(\cdot)$ and $G(\cdot)$. For $D(\cdot) + C(\cdot)$, we train to maximize $-\mathcal{L}_s + \mathcal{L}_d^r - \mathcal{L}_d^f$, while $G(\cdot)$ is trained to maximize $-\mathcal{L}_s + \mathcal{L}_d^f$.

Remark: For $C(\cdot)$, W_c are the classifier parameters for both the base and novel classes. We initialize the classifier parameters trained on the base and novel dataset with the ResNet-34 deep features [125] (See detail in experiments). As known to all, a deep model training on base classes with many samples per class can achieve very promising results for base classes [110]. That is, the classifier parameters are good enough for base classes recognition. The goal of one-shot learning is to improve the classifier parameters for one-shot classes. Hence, we hope the base classifier parameters to be similar to the pre-training one. We develop a square loss regularizer to constrain the base classifier not far away from its initialized one. In this way, not only can we update the classifier parameters for novel classes to enhance the classification ability, but also relax the classifier space for base classes, triggering the expansion of novel classifier space.

Implementation: Training with one-shot classes usually results in a biased classifier. Our algorithm aims to correct this classifier bias by transferring variances from base classes to one-shot classes. For the generative one-shot learning model, we adopt the deep features from (ResNet-34) with loss (Eq. (5.1)) as the input and set the learning rate as 10^{-4} with the optimizer as Adam optimizer. We adopt leaky-relu and sigmoid activation functions for $G(\cdot)$ and $D(\cdot)$, respectively. Since GANs can be solved as a *minimax* optimization problem, we first constrain the generator to optimize the discriminator, then fix the discriminator to update the generator. Thus, we iteratively update two neural networks until the model converges. This violation will help to correct the classifier bias and reshape the decision boundary (Figure 5.3).

5.4 Experimental Results

In this section, we first introduce the one-shot face data as well as its feature representation process. Then we provide the one-shot evaluations with other comparisons to verify the effectiveness of our proposed model. Finally, we go deep and show some phenomena of our model.

We first train a general face representation model with the training images in the base set, and then train a multi-class classification model with the training images in both the base and novel sets. We list the experimental results in details in the following subsections.

5.4.1 One-Shot Face Dataset

The face dataset[5] used here is sampled from MS-Celeb-1M dataset [168]. In total, this dataset contains 21K people with 1.2M images, which is considerably larger than other publicly available datasets except for the MS-Celeb-1M dataset. To evaluate the one-shot challenge, we divide the dataset into base set (20K) parts, i.e., base set (20K people) and novel set (1K people). Since we want to build a general 21K-class classifier for both the base and novel classes, we hope our optimized classifier achieve promising performance on both sets, otherwise it is meaningless in the real-world applications.

In the base set, there are 20K persons, each of which having 50-100 images for training and 5 for test. In the novel set, there are 1000 persons, each with one image for training and 10 for test. The experimental results in this paper were obtained with 100K test images for the base set and 20K test images for the novel set. We focus on the recognition performance in the novel set, while monitoring the recognition performance in the base set to ensure that the performance improvement in the novel set does not harm the performance in the base set.

To recognize the test images for the persons in the novel set is a challenging task. The one training image per person was randomly preselected, and the selected image set includes images of low resolution, profile faces, and faces with occlusions. The training images in the novel set show a large range of variations in gender, race, ethnicity, age, camera quality (or evening drawings), lighting, focus, pose, expressions, and many other parameters.

5.4.2 Face Representation Learning

Learning good feature is the foundation of one-shot face recognition task. In order to evaluate the discrimination and generalization capability of our face representation model, we leverage the LFW [169, 170] verification task, which is to verify whether a given face pair (in total 6000) belongs to the same person or not.

We train our face representation model (Eq. (5.1)) using the images in our base set (already published to facilitate the research in the area, excluding people in LFW by design) with ResNet-34 [125] with input faces' resolution as 224×224. Specifically, we seek a 20K-class classifier using all the training images of the 20K persons in the base set. There are about 50-100 images per person in the base set. The wrong labels in the base set are very limited (less than 1% based on manual

[5] http://www.msceleb.org/challenge2/2017

Table 5.1: LFW verification results obtained with models trained with our published base set. All the models use ResNet-34 [125] as the feature extractor. For the sphere face, please refer to our paper for explanation (fail to converge).

Methods	Network	Accuracy
Cross entropy only	1	98.88%
Center face [128]	1	99.06%
Sphere face [121]	1	$-.--\%$
Cross entropy + WFB in Eq. (5.1) (ours)	1	$\mathbf{99.28}\%$

check). We crop and align face areas to generate the training data[6]. Our face representation model is learned from predicting the 20K classes. We have tried different network structures and adopted the standard residual network with 34 layers [125] due to its good trade-off between prediction accuracy and model complexity. Features extracted from the last pooling layer are adopted as the face representation (512 dimensions).

The verification accuracy with different models are listed in Table 5.1. As shown, for the loss function, we investigated the standard cross entropy, cross entropy plus our WFB-loss term in Eq. (5.1), the center loss in [128], and the sphere face loss in [121]. For the WFB-loss, we set λ in Eq. (5.1) as 0.1. For the center loss, we tried different sets of parameters and found the best performance could be achieved when the balancing coefficient was 0.005, as reported in the table. For the sphere face [121], we noticed this paper very recently and only tried limited sets of parameters (there are four parameters to be adjusted together). The parameters reported in the paper can not make the network converge on our dataset. The only parameter set we found to make the network converge leads to worse results, compared with the standard cross-entropy loss. Due to time constraint, for the other methods, we only report the results for some of them referring the numbers stated in the published corresponding papers. Please note that these methods use different datasets and different networks structures.

As shown in Table 5.1, we obtain the face representation model with the cutting-edge performance with the help of our WFB-loss term in Eq. (5.1). We regard our model good enough to let us start to investigate the one-shot learning phase.

We also tried different values of λ in Eq. (5.1) and found our method is not sensitive to the choose of λ, shown in the Figure 5.5. Larger λ means stronger regularizer applied. Note $\lambda = 0$ corresponds to no WFB-loss applied. From the results, we found $\lambda = 0.1$ could generate better

[6] http://www.msceleb.org/download/lowshot

Table 5.2: For reference, LFW verification results (partially) reported in peer-reviewed publications. Different datasets and network structures were used.

Methods	Dataset	Network	Accuracy
JB [167]	Public	–	96.33%
Human	–	–	97.53%
DeepFace[35]	Public	1	97.27%
DeepID2,3 [171, 172]	Public	200	99.53%
FaceNet [34]	Private	1	99.63%
Center face [128]	Private	1	99.28%
Center face [128]	Public	1	99.05%
Sphere face [121]	Public	1	99.42%
Our WFB in Eq. 5.1	Public	1	**99.73%**

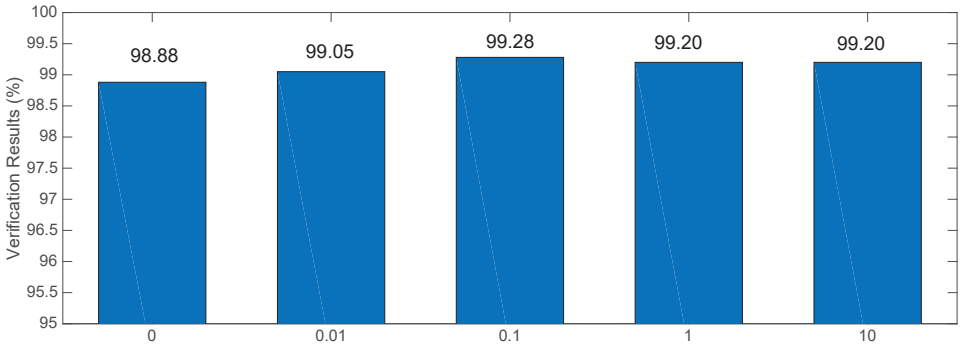


Figure 5.5: LFW verification results obtained with different λ for our WFB in Eq. (5.4), where x-axis denotes the values of λ.

performance.

5.4.3 One-shot Face Recognition

In phase two, we train a $21,000$-class classifier to recognize the persons in both the base set and the one-shot set. Since there is only one image per person for training in the one-shot set, we repeat each sample in the one-shot set for 100 times through all the experiments in this section. In order to test the performance, we apply this classifier with $120,000$ test images consists of images from the base or one-shot set. We focus on the recognition performance in the novel set while monitoring the recognition performance in the base set to ensure that the performance improvement

Table 5.3: Coverage at Precisions = 99% and 99.9% on the one-shot set, where our generative model significantly improves the coverage at precision 99% and 99.9%.

Method	C@P=99%	C@P=99.9%
Fixed-Feature	25.65%	0.89%
SGM [114]	27.23%	4.24%
Update Feature	26.09%	0.97%
Direct Train	15.25%	0.84%
Shrink Norm [110]	32.58%	2.11%
Equal Norm [110]	32.56%	5.18%
Up Term [110]	77.48%	47.53%
Hybrid [111]	92.64%	N/A
Doppelganger [173]	73.86%	N/A
Generation-based [141]	61.21%	N/A
Ours	94.98%	83.94%

in the novel set does not harm the performance in the base set.

To recognize the test images for the persons in the novel set is a challenging task. The one training image per person was randomly preselected, and the selected image set includes images of low resolution, profile faces, and faces with occlusions. We provide more examples in the supplementary materials due to space constraint. The training images in the novel set show a large range of variations in gender, race, ethnicity, age, camera quality (or evening drawings), lighting, focus, pose, expressions, and many other parameters. Moreover, we applied de-duplication algorithms to ensure that the training image is visually different from the test images, and the test images can cover many different looks for a given person.

We compare with the following algorithms:

- **Fixed-Feature**: updates the feature extractor and only train the Softmax classifier with the feature extractor provided by phase one.
- **Updated Feature**: fine-tunes the feature extractor simultaneously when we train the Softmax classifier in phase two. The feature updating does not change the recognizer's performance too much.
- **SGM** [114]: is known as squared gradient magnitude loss, is obtained by updating the feature extractor during phase one using the feature shrinking method.

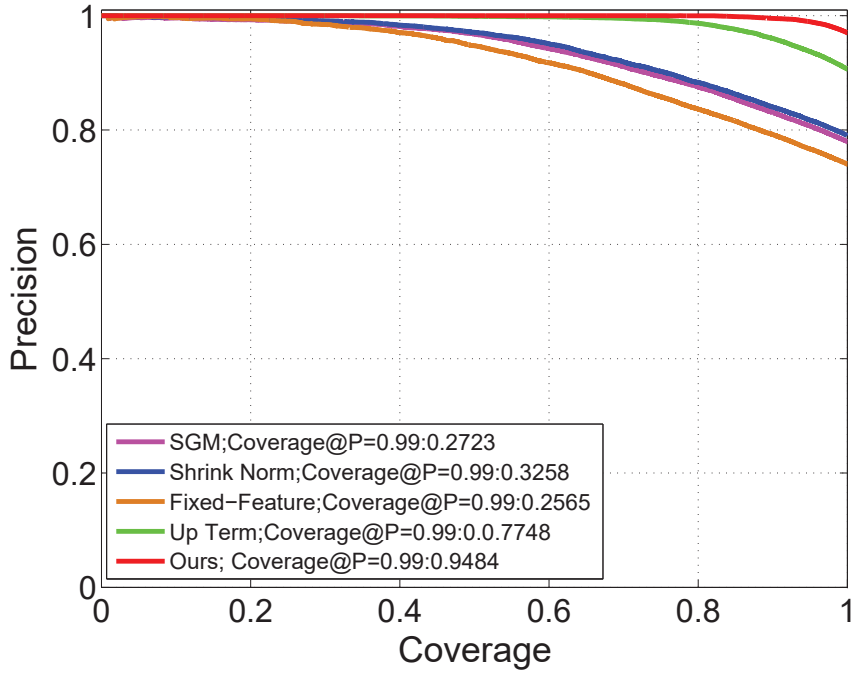


Figure 5.6: Precision-Coverage curves of five methods on the one-shot set, where our model achieves a very appealing coverage@precision=99%.

- **Shrink norm** [110]: adopts L_2-norm to shrink classifier parameters, which is one typical strategy to handle insufficient data problem efficiency.
- **Equal norm** [110]: is a weight regularizer, which constrains the classifier parameters of both novel and base classes to the same value.
- **UP Term** [110]: is a weight regularizer, which only enforces the classifier parameters of the novel classes to the same value.

All the methods are based on a 21K-class classifier (trained with different methods). Note that we boost all the samples in the novel set for 100 times for all the methods, since the largest number of samples per person in the base set is about 100.

The experimental results of our method and the alternative methods are listed in Table 5.3. We adopt coverage rate at precision 99% and 99.9% as our evaluation metrics since this is the major requirement for a real recognizer [110]. As shown in the table, our method significantly improves the recall at precision 99% and 99.9% and achieves the **best** performance among all the methods. Unless numbers reported by other papers (Hybrid, Doppelganger, and Generation-based), the face feature extractor was trained with cross entropy loss.

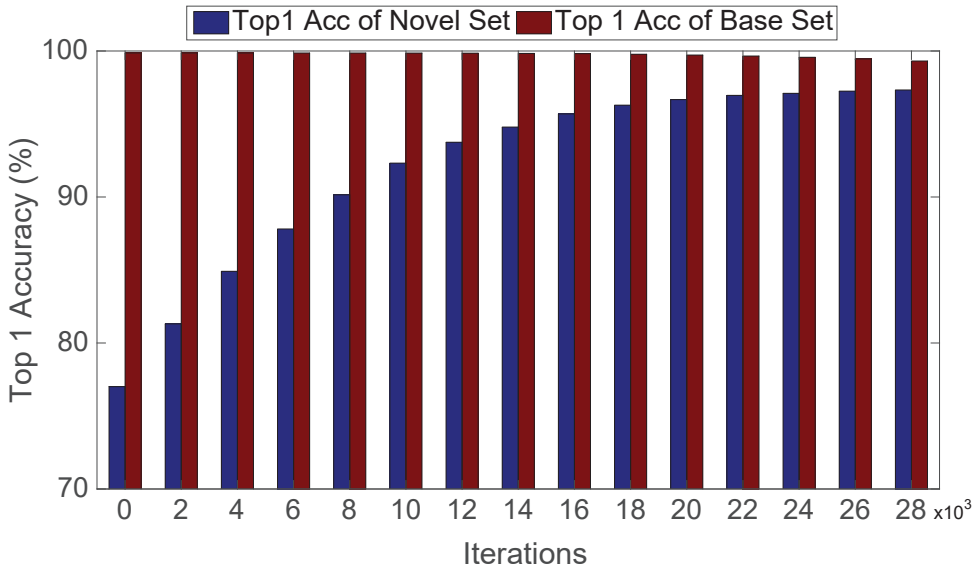


Figure 5.7: Top1 accuracy (%) of base set and novel set with different iterations, where we notice that our model could significantly improve the Top1 accuracy for the novel classes while keeping a very promising Top1 accuracy for the base classes.

Compared with the Fixed-Feature, SGM method obtains around 2% improvements in recall when precision is 99%, while 4% improvements when precision requirement is 99.9%. The gain for face recognition by feature shrinking in [114] is not as significant as that for general image. The reason might be that the face feature is already a good representation for faces and the representation learning is not a major bottleneck. Note that we did not apply the feature hallucinating method as proposed in [114] for fair comparison and to highlight the contribution of model learning, rather than data augmentation. To couple the feature hallucinating method (may need to be modified for face) is a good direction for the next step.

Our model significantly improves the one-shot classification, preserving base classification at a very promising performance. Specifically, as shown in Table 5.3, our generative model improves the coverage@precision=99% and coverage@precision=99.9% significantly. Moreover, we notice that our model can achieve the state-of-the-art performance without any external data by comparing the competitors in the low-shot challenge [7]. This verifies our generative model is able to synthesize very effective features to alleviate the one-shot classification.

The coverage at precision 99% on the base set obtained by using any classifier-based

[7] http://www.msceleb.org/leaderboard/c2

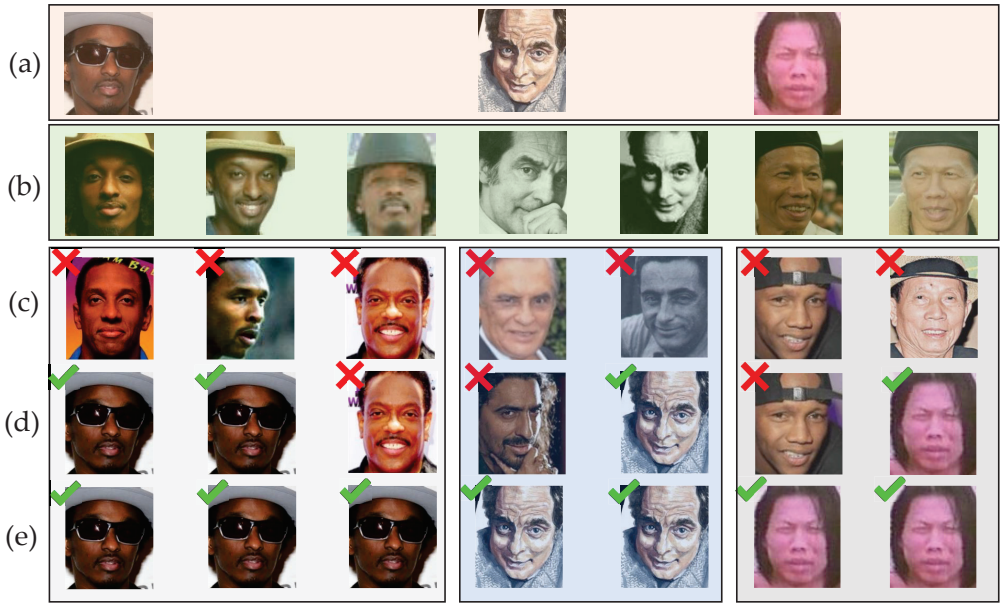


Figure 5.8: Face retrieval results, where row (a) denotes the three challenges one-shot training faces, i.e., occlusion, sketch, low-resolution. Row (b) represents the test images, while the the bottom three rows show the recognized results of three models, i.e., (c) k-NN, (d) Softmax, and (e) Our generative model.

methods in Table 5.3 is 100%. The Top1 accuracy on the base set obtained by any of these classifier-based methods is $99.80 \pm 0.02\%$. Thus, we do not report them separately in the table. That verifies that our generative model could synthesize meaningful one-shot samples to boost classifier space for one-shot classes.

5.4.4 Face Retrieval Results

We select three typical one-shot training cases (Figure 5.8), i.e., low-solution, sketch, occlusion, to quantitatively show the performance of different models. We compare with k-nearest neighbor classifier (k-NN) ($k = 1$), Softmax, and our one-shot generative model. All the models are input with the pre-trained deep ResNet-34 features with our newly designed loss (Eq. (5.1)).

From the results (Figure 5.8), we observe that our model can well handle these three challenging cases and recognize these persons correctly. k-NN cannot correctly recognize the testing images of these three persons, which results from that the testing images are quite different from the one-shot training image. Softmax can retrieve some correct ones, which shows more promising results than k-NN. Hence, we consider KNN is not suitable in one-shot face classification in large-

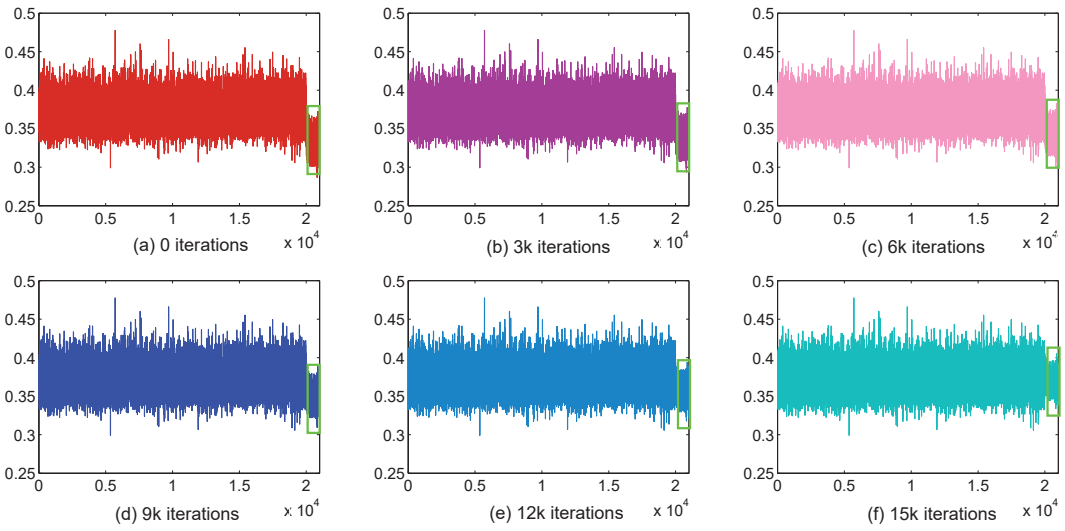


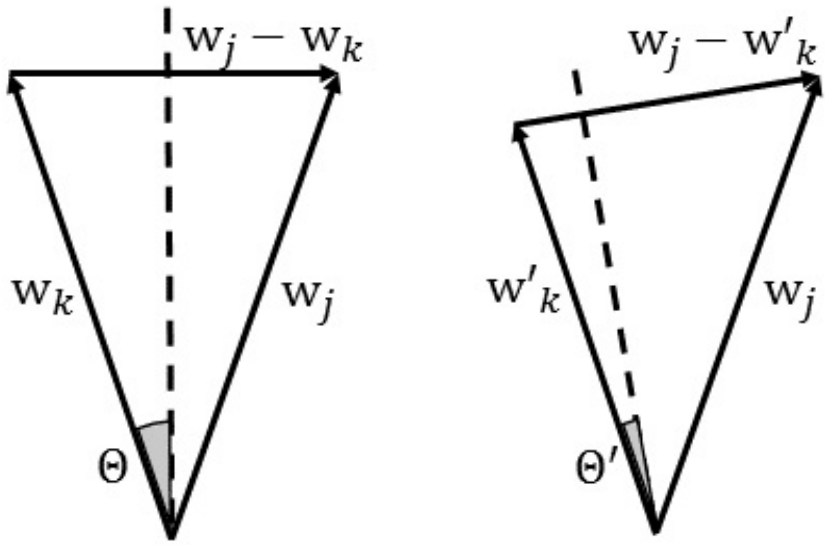
Figure 5.9: Norm of the classifier weight vector w for each class in W_c. The x-axis is the class index. The rightmost 1000 classes on the x-axis correspond to the persons in the novel set. As shown in the figure, with more iterations from (a) to (f), $\|w\|_2$ for the novel set tends to have similar values as that of the base set (Green bounding box denotes the weights for one-shot classes). This promotion introduces significant performance improvement.

scale dataset. Our model can significantly handle those three challenging cases, which results from the generation of effective data in facilitating the classifier learning.

5.4.5 Property Analysis

First of all, we evaluate the Top1 accuracy of the base set and novel set with the model optimization. From the results 5.7, we observe that the Top1 accuracy of one-shot set is significantly improved from 77.01% to 96.82%. This shows that our model enhances the classification for one-shot classes by spanning the feature space. We further notice that the classification accuracy for the base set is hurt somehow, but very slightly. That demonstrates our generative model can learn a good general classifier, which is much practical in real-world scenarios.

Secondly, we present more information for the classifier to deeply understand why our model can improve the one-shot classification. Specifically, we have a c-class classifier, with each class weight vector w. Thus, we evaluate the norm of each class weight vector to see the variations of these information. From the results (Figure 5.9), we notice that from (a) to (f) with more iterations' optimization, the norms of the novel classes are triggered to similar distribution as the base classes



(a) $\|\mathbf{w}_k\|_2 = \|\mathbf{w}_j\|_2$ (b) $\|\mathbf{w}_k\|_2 < \|\mathbf{w}_j\|_2$

Figure 5.10: Relationship between the norm of $\mathbf{w}_k$ and the volume size of the partition for the k-th class. The dash line represents the hyper-plane (perpendicular to $\mathbf{w}_j - \mathbf{w}_k$) which separates the two adjacent classes. As shown, when the norm of $\mathbf{w}_k$ decreases, the k-th class tends to possess a smaller volume size in the feature space.

(f). Actually, (a) shows the results of the initialized parameters obtained from Softmax trained on ResNet-34 deep features. That is the reason we consider why our model significantly improves the one-shot classification, since we boost the mean of novel classes to be similar to base classes. Such phenomenon is also obtained in [110], where the assume the norm of classifier weight is related to the classifier space. To further understand this property, without loss of generality, we discuss the decision hyperplane between any two adjacent classes. Note we set all the bias terms b_k and b_j to 0 throughout the paper. With this setup, we apply Eq. (5.3) to both the k-th class and the j-th class to determine the decision hyperplane between the two classes (note we do not have bias terms throughout our paper):

$$\frac{p_j(x)}{p_k(x)} = \frac{\exp(\mathbf{w}_j^\top \boldsymbol{\phi}(x))}{\exp(\mathbf{w}_k^\top \boldsymbol{\phi}(x))} = \exp[(\mathbf{w}_j - \mathbf{w}_k)^\top \boldsymbol{\phi}(x)] \tag{5.15}$$

As shown in Figure 5.10, the hyperplane to separate two adjacent classes k and j is perpendicular to the vector $\mathbf{w}_j - \mathbf{w}_k$. When the norm of $\mathbf{w}_k$ gets decreased, this hyperplane is pushed towards the k-th class, and the volume for the k-th class also gets decreased. As this property holds

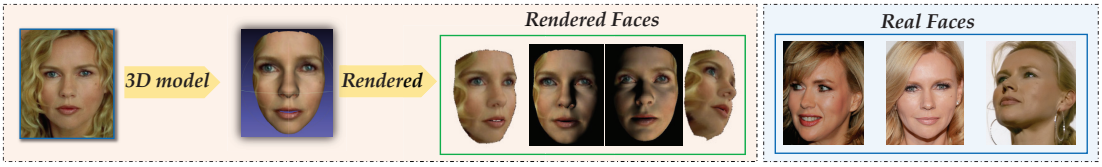


Figure 5.11: Illustration of 3D face reconstruction based on a single image, where we adopt the 3D model [174] to build the 3D face, then render different images based on different poses. We could see the rendered faces are still much different from the real faces of that person.

for any two classes, we can clearly see the connection of the norm of a weight vector and the volume size of its corresponding partition space in the feature space.

5.5 Future Direction of One-Shot Learning

One-shot learning is a very important and practical problem in the real-world, which will continuously attract attentions in the literature. Considering the promising achievements in the one-shot face recognition in traditional image domain, we post two directions of one-shot learning in the future work, one is *Joint 3D Reconstruction & One-shot Learning*, and the other is *Cross-modal One-shot Face Recognition*.

5.5.1 Joint 3D Reconstruction & One-shot Learning

Beyond face recognition, it is also a very essential topic to reconstruct the 3D face model from a set of input such as image(s), video, or depth data. It is a difficult problem with much recent interest and a variety of applications. In the biometrics community, pose, expression, and illumination are the main challenges of face recognition and all may be improved with accurate person-specific face models [175, 176]. In graphics, high fidelity models with skeletal structures are useful for animations, puppeteering, and post processing videos.

Recently, great research efforts are exploited on reconstructing faces from photo collections [177, 178, 174]. Along this line, the seminal work creates a 2.5D model, locally consistent with the photo collection [177], which is extended in a few different directions, one is to use the surface normals from frontal faces to improve the fitting of a 3D Morphable Model (3DMM)[8], two is to generate a 3DMM, and three to handle pose variation and reconstructs a 3D model [178]. Later on,

[8] http://cvssp.org/faceweb/3dmm/

Roth et al. continues to improve the 3D reconstruction technique by adapting lower-quality photo collections with fewer input images [174].

Most recently, 3D face reconstruction on single face attracts great interests. Specifically, Jackson et al. proposed a Direct Volumetric CNN Regression, aiming to reconstruct the whole 3D facial geometry (including the non-visible parts of the face) bypassing the construction (during training) and fitting (during testing) of a 3DMM [179]. Richardson et al. proposed a two-block architecture, a network that recovers the coarse facial geometry (CoarseNet), followed by a CNN that refines the facial features of that geometry (FineNet) [180].

However, current 3D face reconstruction on a single image or even a collection of images is far from satisfactory (Figure 5.11). It is still a long to go when obtaining promising results. On the other hand, it is a natural strategy through rendering face images to augment the one-shot classes. While if the 3D model is not that accurate, the rendered face images may not help a lot. Based on the generative one-shot learning model proposed in this work, it is an appealing topic that we could joint 3D face recognition and one-shot face recognition together. The idea is to synthesize more meaningful faces based on the rendered one, then we could further optimize the 3D face model.

5.5.2 Cross-modal One-shot Face Recognition

Current one-shot face recognition mainly focuses on the base classes and one-shot classes sampled from similar or same distributions, i.e, they are from the same RGB image domain. However, we may confront multi-modal face images in real-world applications, e.g., forensic face recognition. In this scenario, we may have a large set of base classes from traditional RGB domain while one-shot classes from other domains, e.g., sketch domain, near-infrared domain, even short/middle/long wave infrared domain. Here we propose two directions for cross-modal one-shot face recognition.

First of all, taking the forensic face recognition as an example, we have a sketch face as reference to search RGB faces through surveillance camera. In this problem, we could have a large-scale public face dataset as source domain to help the task. That is, we need to train a model on base and one-shot classes with different distributions, which is the key difference with the existing one-shot face recognition.

Secondly, how can we adapt the knowledge from current large-scale public face datasets (RGB domain) to face recognition in new domains, where we only have one training sample? This is different from the above one, that is the test images are still sampled from the one-shot domain. For example, we aim to develop a face system well handling night-time, low illumination environments.

In a challenging case, we only have limited training samples even one training sample in that environment, how can we still borrow the knowledge from public large-scale face datasets? It can be treated as *one-shot transfer learning*, since in the target domain we are only accessible to one sample per class.

5.6 Conclusions

In this chapter, we first presented a comprehensive review of one-shot face recognition, including what is one-shot learning problem, what is the challenge of one-shot learning, and what is the current research status for one-shot learning. We believe this paper could provide readers a systematical understanding of one-shot learning problem and benefit the computer vision community in both industry and academia from literature review to future directions, i.e., *Joint 3D Reconstruction & One-shot Learning*, and *Cross-modal One-shot Face Recognition*.

More importantly, we proposed a generative framework for one-shot face recognition, where we attempted to synthesize more effective augmented data for one-shot classes by borrowing the data variation of base set. Specifically, generative learning was jointly incorporated in the general classifier training for both the base and novel classes. Thus, more effective fake data were generated for the one-shot classes to enrich the data space of one-shot classes. Furthermore, a discriminator was designed to guide the face data generation to mimic the data variation of base classes and adapt to generate novel classes. Experiments on a large-scale one-shot face benchmark showed that our model could significantly improve the performance of one-shot classification, while keeping the promising classification ability for the base set.

Chapter 6

Conclusion

Forensic face recognition is a very important technique in forensic science, whose major issue is the unstable system performance due to internal factor, e.g., aging, and external factors, e.g., image resolution/modality, illumination, pose. In this thesis, we consider several challenges in forensic face recognition, e.g., multi-view face recognition with view-unknown test data, missing modality face recognition, and one-shot face recognition.

In chapter 2, we develop a view-invariant framework based on both subspace and deep learning to address the test multi-view data with view information unknown. Specifically, we explore multiple view-specific structures and one view-invariant structure to solve this issue.

In chapter 3, we mainly explore to utilize the knowledge of one domain to do face recognition on another domain. We consider the missing modality problem, where we have no test data available in the training stage. Specifically, we design a two-directional transfer learning framework to iteratively seek a domain-invariant feature space.

In chapter 4, we develop a deep feature learning framework aiming to seek better feature representation for face recognition. Specifically, we build a deep auto-encoder architecture by constraining the output of the decoded features to a low-rank basis, so that our designed deep architecture is able to well deal with noisy data.

In chapter 5, we target at solving one-shot face recognition, where only one training sample is available for some persons in the training stage. We first systematically review the literature of one-shot learning, then we develop a generative one-shot face recognition model by synthesizing more valid data for one-shot persons.

Bibliography

[1] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," in *Proceedings of European Conference on Computer Vision*. Springer, 2012, pp. 808–821.

[2] Z. Ding and Y. Fu, "Low-rank common subspace for multi-view learning," in *IEEE International Conference on Data Mining*, 2014.

[3] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in *IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 3176–3183.

[4] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[5] X. Wu and Y. Jia, "View-invariant action recognition using latent kernelized structural svm," in *European Conference on Computer Vision*. Springer, 2012, pp. 411–424.

[6] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," *arXiv preprint arXiv:1404.3543*, 2014.

[7] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked progressive auto-encoders (spae) for face recognition across poses," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1883–1890.

[8] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: a deep model for learning face identity and view representations," in *Advances in Neural Information Processing Systems*, 2014, pp. 217–225.

[9] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[10] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *European Conference on Computer Vision*. Springer, 2012, pp. 702–715.

[11] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[12] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars *et al.*, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013.

[13] S. Shekhar, V. Patel, H. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[14] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 3453–3459.

[15] ——, "Latent low-rank transfer subspace learning for missing modality recognition," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[16] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, pp. 1–20, 2014.

[17] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 325–333.

[18] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 692–699.

[19] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[20] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 504–518, 2012.

[21] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa, "Dash-n: Joint hierarchical domain adaptation and feature learning," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5479–5491, 2015.

[22] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 44–51.

[25] A. Droniou and O. Sigaud, "Gated autoencoders with tied input weights," in *International Conference on Machine Learning*, 2013, pp. 154–162.

[26] M. Kan, S. Shan, and X. Chen, "Bi-shifting auto-encoder for unsupervised domain adaptation," in *IEEE International Conference on Computer Vision*, 2015, pp. 3846–3854.

[27] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," *IEEE International Conference on Computer Vision*, 2015.

[28] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1083–1092.

[29] C. Xia, F. Qi, and G. Shi, "Bottom–up visual saliency estimation with deep autoencoder-based sparse reconstruction," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 6, pp. 1227–1240, 2016.

[30] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[31] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[32] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, June 2014.

[33] ——, "Web-scale training for face identification." in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 2746–2754.

[34] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[35] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[36] M. Shao and Y. Fu, "Cross-modality feature learning through generic hierarchical hyperlingual-words," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2016.

[37] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Regularized latent least square regression for cross pose face recognition," in *Twenty-Third international joint conference on Artificial Intelligence*, 2013, pp. 1247–1253.

[38] M. Du, A. Sankaranarayanan, and R. Chellappa, "Robust face recognition from multi-view videos," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1105–1117, March 2014.

[39] Z. Ding and Y. Fu, "Robust multi-view subspace learning through dual low-rank decompositions," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1181–1187.

[40] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, "Unsupervised metric fusion over multi-view data by graph random walk-based cross-view diffusion." *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[41] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3939–3949, 2015.

[42] L. Niu, W. Li, D. Xu, and J. Cai, "An exemplar-based multi-view domain generalization framework for visual recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[43] W. Yang, Y. Gao, Y. Shi, and L. Cao, "Mrm-lasso: A sparse multiview feature selection method via low-rank analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2801–2815, 2015.

[44] X. Huang, Z. Lei, M. Fan, X. Wang, and S. Z. Li, "Regularized discriminative spectral regression method for heterogeneous face matching," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 353–362, 2013.

[45] H. Hotelling, "Relations between two sets of variates," *Biometrika*, pp. 321–377, 1936.

[46] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Conference on Data Mining and Data Warehouses*, 2010, pp. 1–4.

[47] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[48] Z. Ding, M. Shao, and Y. Fu, "Incomplete multisource transfer learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[49] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 663–670.

[50] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[51] S. Li and Y. Fu, "Learning robust and discriminative subspace with low-rank constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2015.

[52] K. Meina, S. Shiguang, Z. Haihong, L. Shihong, and C. Xilin, "Multi-view discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[53] S. Li and Y. Fu, "Robust subspace discovery through supervised low-rank constraints," in *SIAM International Conference on Data Mining*, 2014, pp. 163–171.

[54] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[55] 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*, vol. 29, no. 1, pp. 40–51, 2007.

[56] Y.-F. Guo, S.-J. Li, J.-Y. Yang, T.-T. Shu, and L.-D. Wu, "A generalized foley–sammon transform based on generalized fisher discriminant criterion and its application to face recognition," *Pattern Recognition Letters*, vol. 24, no. 1, pp. 147–158, 2003.

[57] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[58] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.

[59] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer Science & Business Media, 2006, vol. 1.

[60] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[61] D. L. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[62] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *arXiv preprint arXiv:1511.06324*, 2015.

[63] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[64] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[65] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[66] X. He and P. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16, 2004, p. 153.

[67] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3025–3032.

[68] R. Liu, Z. Lin, F. De la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 598–605.

[69] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database of human faces," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-01-02, January 2001.

[70] H. Di, S. Jia, and W. Yunhong, "The buaa-visnir face database instructions," in *IRIP-TR-12-FR-001*, 2012.

[71] S. Si, D. Tao, and B. Geng, "Bregman divergence -based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.

[72] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *IEEE International Conference on Computer Vision*, 2013.

[73] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–171.

[74] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 1657–1664.

[75] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 10–18.

[76] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *European Conference on Computer Vision*. Springer, 2014, pp. 628–643.

[77] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[78] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[79] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[80] M. Long, J. Wang, G. Ding, J. Sun, and P. Yu, "Transfer joint matching for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1410–1417.

[81] M. Long, J. Wang, G. Ding, S. Pan, and P. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[82] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2168–2175.

[83] B. F. Klare and A. K. Jain, "Heterogeneous face recognition using kernel prototype similarities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1410–1422, 2013.

[84] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 513–520.

[85] W. Li, L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation." *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[86] S. Wang, L. Zhang, L. Y., and Q. Pan, "Semi-coupled dictionary learning with applications in image super-resolution and photo-sketch synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.

[87] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *IEEE International Conference on Computer Vision*, 2011, pp. 1615–1622.

[88] Z. Ding and Y. Fu, "Low-rank common subspace for multi-view learning," in *IEEE International Conference on Data Mining*. IEEE, 2014, pp. 110–119.

[89] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2006, pp. 513–520.

[90] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[91] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009.

[92] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1324.

[93] A. S. Lewis and J. Malick, "Alternating projections on manifolds," *Mathematics of Operations Research*, vol. 33, no. 1, pp. 216–234, 2008.

[94] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 33–40.

[95] M. Shao, C. Castillo, Z. Gu, and Y. Fu, "Low-rank transfer subspace learning," in *IEEE 12th International Conference on Data Mining*, 2012, pp. 1104–1109.

[96] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 647–655.

[97] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Neural Information Processing Systems*, 2013, pp. 2553–2561.

[98] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1701–1708.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[100] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[101] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *International Conference on Machine Learning*, 2011, pp. 265–272.

[102] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *International Conference on Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[103] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in neural information processing systems*, 2009, pp. 2080–2088.

[104] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[105] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 74–93, 2014.

[106] I.-H. Jhuo, D. Liu, D. Lee, S.-F. Chang *et al.*, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2168–2175.

[107] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2586–2593.

[108] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[109] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[110] Y. Guo and L. Zhang, "One-shot face recognition by promoting underrepresented classes," *arXiv preprint arXiv:1707.05574*, 2017.

[111] Y. Wu, H. Liu, and Y. Fu, "Low-shot face recognition with hybrid classifiers," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[112] Y. Wu, J. Li, Y. Kong, and Y. Fu, "Deep convolutional neural network with independent softmax for large scale face recognition," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1063–1067.

[113] I. Kemelmacher-Shlizerman, S. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," *ArXiv e-prints*, 2015.

[114] B. Hariharan and R. Girshick, "Low-shot visual object recognition," *arXiv preprint arXiv:1606.02819*, 2016.

[115] Y.-X. Wang and M. Hebert, "Learning from small sample sets by combining unsupervised meta-training with cnns," in *Advances in Neural Information Processing Systems*, 2016, pp. 244–252.

[116] A. Mehrotra and A. Dukkipati, "Generative adversarial residual pairwise networks for one shot learning," *arXiv preprint arXiv:1703.08033*, 2017.

[117] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tail," in *Proc. of Int'l Conf. on Computer Vision*, 2017.

[118] H. Hanselmann, S. Yan, and H. Ney, "Deep fisher faces," 2017.

[119] Y. Wu, H. Liu, J. Li, and Y. Fu, "Deep face recognition with center invariant loss," in *ACM Multimedia Workshop*, 2017.

[120] J. Deng, J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[121] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. abs/1704.08063, 2017.

[122] H. Edwards and A. Storkey, "Towards a neural statistician," *arXiv preprint arXiv:1606.02185*, 2016.

[123] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[124] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[125] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[126] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[127] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, vol. 3, no. 6, 2017, p. 7.

[128] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[129] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 672–679.

[130] L. Pemula, "Low-shot visual recognition," 2016, master Thesis at Virginia Polytechnic Institute and State University.

[131] Y. Xu, Y. Cheng, J. Zhao, Z. Wang, L. Xiong, K. Jayashree, H. Tamura, T. Kagaya, S. Shen, S. Pranata, J. Feng, and J. Xing, "High performance large scale face recognition with multi-cognition softmax and feature retrieval," in *Proc. of Int'l Conf. on Computer Vision Workshop (ICCV-W)*, Oct 2017.

[132] L. Fei-Fei, F. Rob, and P. Pietro, "A bayesian approach to unsupervised one-shot learning of object categories," in *IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 1134–1141.

[133] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 489–496.

[134] E. Rodner and J. Denzler, "One-shot learning of object categories using dependent gaussian processes," in *Joint Pattern Recognition Symposium*. Springer, 2010, pp. 232–241.

[135] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[136] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Advances in neural information processing systems*, 2013, pp. 2526–2534.

[137] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 928–941, 2014.

[138] Y.-X. Wang and M. Hebert, "Learning to learn: Model regression networks for easy small sample learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 616–634.

[139] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[140] D. Rezende, I. Danihelka, K. Gregor, D. Wierstra *et al.*, "One-shot generalization in deep generative models," in *International Conference on Machine Learning*, 2016, pp. 1521–1529.

[141] J. Choe, S. Park, K. Kim, J. H. Park, D. Kim, and H. Shim, "Face generation for low-shot learning using generative adversarial networks," in *Proc. of Int'l Conf. on Computer Vision Workshop (ICCV-W)*, 2017.

[142] Z. Ding, Y. Guo, L. Zhang, and Y. Fu, "One-shot face recognition via generative learning," in *IEEE International Conference on Automatic Face and Gesture Recognition*, vol. 1. IEEE, 2018, pp. 1–6.

[143] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proceedings of the Advances in neural information processing systems*, 2009, pp. 1410–1418.

[144] D. Parikh and K. Grauman, "Relative attributes," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 503–510.

[145] X. Yu and Y. Aloimonos, "Attribute-based transfer learning for object categorization with zero/one training example," in *Proceedings of the European conference on computer vision*. Springer, 2010, pp. 127–140.

[146] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927–2936.

[147] P. Peng, Y. Tian, T. Xiang, Y. Wang, and T. Huang, "Joint learning of semantic and latent attributes," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 336–353.

[148] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proceedings of the Advances in neural information processing systems*, 2013, pp. 935–943.

[149] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2452–2460.

[150] X. Li, Y. Guo, and D. Schuurmans, "Semi-supervised zero-shot classification with label representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4211–4219.

[151] M. Bucher, S. Herbin, and F. Jurie, "Improving semantic embedding consistency by metric learning for zero-shot classiffication," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 730–746.

[152] X. Xu, T. M. Hospedales, and S. Gong, "Multi-task zero-shot action recognition with prioritised data augmentation," in *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 343–359.

[153] R. Qiao, L. Liu, C. Shen, and A. v. d. Hengel, "Less is more: zero-shot learning from online textual documents with noise suppression," *Proceedings of the IEEE International Conference on Computer Vision*, 2016.

[154] Z. Ding, M. Shao, and Y. Fu, "Low-rank embedded ensemble semantic dictionary for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2050–2058.

[155] X. Xu, F. Shen, Y. Yang, D. Zhang, H. T. Shen, and J. Song, "Matrix tri-factorization with manifold regularizations for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3798–3807.

[156] H. Jiang, R. Wang, S. Shan, Y. Yang, and X. Chen, "Learning discriminative latent attributes for zero-shot classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4223–4232.

[157] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang, "Zero-shot recognition using dual visual-semantic mapping paths," in *Proceedings of the IEEE conference on Computer vision and pattern recognition*. IEEE, 2017, pp. 3279–3287.

[158] Y. Long, L. Liu, F. Shen, L. Shao, and X. Li, "Zero-shot learning using synthesised unseen visual data with diffusion regularisation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[159] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 513–520.

[160] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," *arXiv preprint arXiv:1605.06636*, 2016.

[161] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," *arXiv preprint arXiv:1606.05328*, 2016.

[162] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[163] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[164] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[165] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[166] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 3387–3395.

[167] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *IEEE International Conference on Computer Vision*. IEEE, 2013, pp. 3208–3215.

[168] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 87–102.

[169] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[170] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[171] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, ser. NIPS'14. MIT Press, 2014, pp. 1988–1996.

[172] Y. Sun, X. Wang, and X. Tang, "DeepID3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2014.

[173] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, and G. Lavrentyeva, "Doppelganger mining for face representation learning," in *Proc. of Int'l Conf. on Computer Vision Workshop (ICCV-W)*, 2017.

[174] J. Roth, Y. Tong, and X. Liu, "Adaptive 3d face reconstruction from unconstrained photo collections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4197–4206.

[175] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 787–796.

[176] D. Zeng, Q. Zhao, S. Long, and J. Li, "Examplar coherent 3d face reconstruction from forensic mugshot database," *Image and Vision Computing*, vol. 58, pp. 193–203, 2017.

[177] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1746–1753.

[178] J. Roth, Y. Tong, and X. Liu, "Unconstrained 3d face reconstruction," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 2606–2615.

[179] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3d face reconstruction from a single image via direct volumetric cnn regression," in *IEEE International Conference on Computer Vision*. IEEE, 2017, pp. 1031–1039.

[180] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, "Learning detailed face reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5553–5562.