REPRESENTATION LEARNING AND IMAGE SYNTHESIS FOR DEEP FACE RECOGNITION

By

Xi Yin

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science — Doctor of Philosophy

2018

ABSTRACT

REPRESENTATION LEARNING AND IMAGE SYNTHESIS FOR DEEP FACE RECOGNITION

By

Xi Yin

Face recognition has been advanced a lot in recent years thanks to the development of deep neural networks. The large intra-class variations in pose, illumination, and expression are the long-standing challenges. Learning a discriminative representation that is robust to these variations is the key. In the scenarios of profile pose or long-tail training data, image or feature-level data augmentation is needed. This dissertation presents three different methods to solve these problems. First, we explore a multi-task Convolutional Neural Network (CNN) that aims to leverage side tasks to improve representation learning. A pose-directed multi-task CNN is introduced to better handle pose variation. The proposed framework is effective in pose-invariant face recognition. Second, we propose a Face Frontalization-Generative Adversarial Network (FF-GAN) that can generate a frontal face even from an input image with extreme profile pose. FF-GAN handles pose variation from the perspective of image-level data augmentation. Multiple loss functions are proposed to achieve large-pose face frontalization. The proposed approach is evaluated on various tasks including face reconstruction, landmark detection, face frontalization, and face recognition. Third, a feature transfer learning method is presented to solve the problem of insufficient intra-class variation via feature-level data augmentation. A Gaussian prior is assumed across all the regular classes and the variance are transferred from regular classes to long-tail classes. Further, an alternating training regimen is proposed to simultaneously achieve less biased decision boundaries and more discriminative representations. Extensive experiments have demonstrated the effectiveness of the proposed feature transfer framework.

ACKNOWLEDGMENTS

This dissertation would not have been made possible without the help of many people.

I am very honored to have Dr. Xiaoming Liu as my advisor. His expectation and encouragement have made me achieve more than I could ever have imagined. The time we spent to debug codes, brainstorm, and polish papers has refined my skills in critical thinking, presentation and writing. By setting himself as an example, he has taught me what a good researcher should be like.

It is my great pleasure to have Dr. Anil K. Jain, Dr. Arun Ross, and Dr. Daniel Morris in my Ph.D. guidance committee. As a world-leading researcher, Dr. Jain has inspired many younger generations including me to pursue a Ph.D. Dr. Ross's patience and insightful comments at all presentations have shown me that every researcher desires to be heard. Thanks to Dr. Morris for his attention to the details and valuable insights in our collaboration.

I would like to thank Dr. Xiang Yu, Dr. Kihyuk Sohn, and Dr. Manmohan Chandraker at NEC Labs for offering me an internship and the collaboration opportunity.

I am grateful for my labmates, Joseph Roth, Amin Jourabloo, Morteza Safdarnejad, Yousef Atoum, Luan Tran, Garrick Brazil, Yaojie Liu, Bangjie Yin, Joel Stehouwer, Adam Terwilliger. The valuable comments in paper review, the willingness to help, the encouragement when I am in a bad mood, and the entertainment together have made it a very pleasant journey.

I am lucky to have met Sandy and Mick who have treated me like their daughter, which means a lot to me. Thanks to my friends at Michigan State University, Huiyun, Wei, Sahra, Panpan, Chelsea, Kailyn, Jiao, Shaohua, Sorrachai for their company to keep my mind refreshed.

Finally, I would like to thank my parents who have taught me to be brave, positive, and kind-hearted. Thanks to my friends Dan, Zi, Bei, Renjie for the previous friendship since childhood. Thanks to Dr. Yu Wang for his intelligence and humor that make my life colorful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction and Contributions

1.1 Face Recognition

Face recognition is one of the most studied and long-standing research topics in computer vision. The key of face recognition is to extract discriminative identity features, or representation, from a face image. The first face recognition system [170] dates back to 1960s, when manual measurement of facial shape is used as the features to identify subject. Later work focus on hand-crafted features such as LBP [2], HOG [34], SIFT [91], and controlled face images that are collected in the lab environment with cooperative subjects [45, 108]. In the last decade, in-the-wild face recognition is the main focus. With the development of neural networks [78, 56, 61], deep features [115, 33, 139] have achieved impressive performance on several challenging face recognition benchmarks [62, 77, 116, 75], where Labeled Faces in-the-Wild (LFW) [62] has been extensively evaluated on until surpassing human performance is achieved [115].

Deep Neural Networks (DNNs) [78] have been successfully applied to many vision applications including face recognition [101, 115, 132], pedestrian detection [131, 158, 13], semantic segmentation [90, 98, 85], and etc. This is attributed to the development of advanced network structures [127, 56, 61, 60], large labeled training data [49, 30], and powerful computing resources. Moreover, DNN frameworks make it possible for end-to-end learning, i.e., to learn a mapping from the input image space to the target label space. In the concept of face recognition, previous

Figure 1.1: Examples of face images showing the challenges of face recognition caused by pose, expression, occlusion, illumination, and image blur.

methods typically consist of two steps: high-dimensional feature extraction [19] and discriminative subspace learning [18]. With DNNs, the above two steps can be combined into a unified framework. Such a formulation is preferable because the loss function can directly supervise the representation learning process. How to design novel loss functions or new structures to learn a better representation is an essential topic that has been widely studied [132, 51, 89].

We study representation learning and image synthesis for deep face recognition, which is a challenging problem due to the large intra-class variations including Pose, Illumination and Expression (collected denoted as PIE), as well as occlusion and image blurring (as shown in Figure 1.1). General deep face recognition algorithms handle these challenges implicitly, i.e., without considering the source of variations, by imposing large inter-class distance and small intra-class distance regularizations [115, 141, 160]. On contrary, supervised learning with auxiliary labels can be employed to study face recognition robust to specific variations [151, 71], among which pose has been considered as the largest challenge. Pose-Invariant Face Recognition (PIFR) has been studied a lot. A comprehensive survey on PIFR can be find at [36]. Existing algorithms can be mainly classified into learning-based and synthesis-based methods. Learning-based methods [153, 132] aim to design a novel loss function or framework to learn more discriminative features from a profile input face. Synthesis-based methods [154, 63] target at generating a frontal-view face image of the same subject, which is easier for face recognition. Representation learning and image synthesis

can be combined in a unified framework [151, 132].

In this dissertation, we first introduce a multi-task Convolutional Neural Network (CNN) with face recognition as the main task and PIE estimations as the side tasks to learn PIE-invariant representation. It may sound straight-forward and intuitive to have PIE labels as auxiliary supervisions, but we are actually the first to do so. We prove that the side tasks regularize the network to disentangle the PIE variations from the learnt identity representation. We overcome the problem of loss weight selection in multi-task learning by formulating a dynamic-weight scheme to learn the weights in the CNN framework. To better handle pose variation, we further propose a pose-directed multi-task CNN to learn pose-specific identity features for face images with different pose groups. Such frameworks are extended to in-the-wild face recognition where no ground truth PIE labels are available. Instead, we use the estimated pose labels as the soft labels for training, which is shown to be effective than the single-task learning framework.

Besides representation learning-based methods for PIFR, image synthesis-based methods [154, 63, 8] are attracting more attentions lately, owing to the success of Generative Adversarial Network (GAN) [44] and its variants [96, 23, 46]. Goodfellow et al. introduce GAN to learn generative models via an adversarial training process. It consists of a generator and a discriminator that improve themselves with a minimax two-player game. GAN has improved dramatically since it was proposed. As shown in Figure 1.2, the original GAN can only generate low quality face images. Recently, Karras et al. [74] propose to progressively train GAN from coarse to fine in order to improve the image resolution up to 1024×1024. While the generated images can be visually appealing, it is more important to enforce identity preservation in the generated images so that it can better contribute to face recognition.

As the second work of this dissertation, we propose Face Frontalization GAN (FF-GAN) to tackle PIFR from the perspective of face image synthesis. We aim to generate an identity-preserved

Figure 1.2: Examples of face images generated by GAN. Top row shows the images generated by [44] where the face resolution is less than 100×100. Bottom row shows the images generated by [74] where the face resolution is as large as 1024×1024.

frontal face from an input face image with arbitrary view even up to extreme profile. The generated face images can help to boost the face recognition performance. There are two challenges in large-pose face frontalization. First, frontal-profile face pairs are limited for deep neural network training. Second, preserving identity in the generated faces is difficult. To solve these issues, we utilize the 3D Morphable Models [12] to provide shape and texture priors into the CNN framework to accelerate training with limited data. Multiple loss functions including symmetry loss and recognition loss are proposed to ensure identity preservation. During the testing stage, we fuse the features of the generated image and the original image for face recognition. FF-GAN can be considered as an image-level data augmentation method for PIFR.

From simple image transformation like scale, rotation, noise etc., to generative models such as GAN and VAE [76], image-level data augmentation techniques are widely accepted and very popular. Meanwhile, feature-level data augmentation has attracted relatively less attention. Until recently, Cao et al. [15] propose an equivariant mapping method to learn a residual to map a profile face to a frontal one in the feature space. Such a feature-level data augmentation is effective for

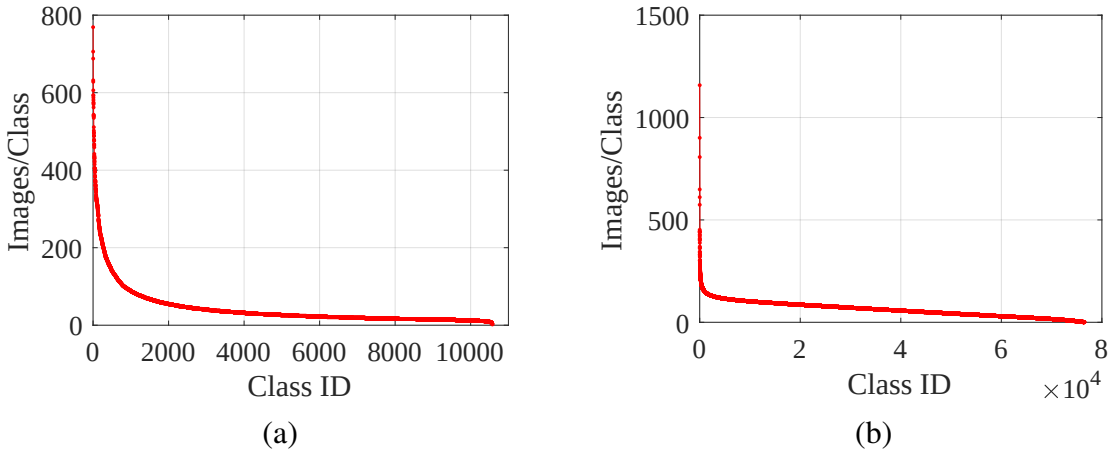


Figure 1.3: Dataset distributions for (a) CASIA-Webface and (b) MS-celeb-1M.

PIFR. Tan et al. [130] introduce a new technique named Feature Super-Resolution (FSR). FSR aims to improve the discriminatory power of the low-resolution images by performing super-resolution in the feature space. These feature-level transformations are easier for identity preservation compared to image-level face generation.

As the last part of this dissertation, we propose a feature transfer learning framework to perform feature-level data augmentation for long-tail classes during the training of a face recognition system. State-of-the-art face recognition algorithms have benefited from increasing volume of training data [150, 49, 16], mostly in the data breath (number of different identities) rather than depth (number of images per identity). In fact, most face recognition datasets are long-tailed. Figure 1.3 shows the dataset distribution of two most popular datasets CASIA-Webface [150] and MS-celeb-1M [49] where most subjects are with limited number of images. Training with long-tailed dataset will result in biased classifier and degraded representation, as observed in [48, 160]. Our proposed framework aims to alleviate this problem by enriching the data distribution of the long-tail classes via center-based feature transfer. An alternative training strategy is introduced to achieve this goal. Both quantitative and qualitative results have validated that the proposed method can indeed augment the feature space of long-tail classes and thus learn a better representation.

Our framework can be generalized to low-shot object recognition.

1.2 Dissertation Contributions

This dissertation studies unconstrained face recognition using deep neural networks. Specifically, we explore three problems, i.e., pose-invariant face recognition, large-pose face frontalization, and training face recognition framework with long-tail data. This dissertation makes the following contributions in order to solve the above problems.

- A multi-task Convolutional Neural Network (m-CNN) is formulated to leverage side tasks for improved representation learning. A dynamic weighting technique is proposed that can bypass the tedious loss weight search in prior multi-task learning frameworks. We provide insights on how side tasks can help to disentangle the variations for robust face recognition.

- A pose-directed multi-task CNN is introduced on top of m-CNN by applying divide-and-conquer into the CNN framework. A stochastic routing scheme is proposed to calculate reliable distance measurement that is robust to potential pose estimation errors during the testing stage. This approach is applicable to other modality-aware recognition tasks.

- A GAN-based end-to-end framework is proposed to achieve face frontalization even for extreme profile faces. We are the first to combine 3DMM into the deep learning framework for face frontalization. Effective loss functions including symmetry loss and smoothness regularization are introduced to lead the generation of high-quality images.

- The proposed FF-GAN can be used for various tasks including 3D reconstruction, landmark localization, face frontalization, and face recognition. State-of-the-art face frontalization and recognition results are achieved on multiple datasets.

- The problem of training face recognition system with long-tailed data is identified and addressed via feature transfer learning. A novel framework is proposed to allow feature transfer and recognition at different feature spaces. The proposed framework allows the visualization of facial features, providing insights on what have been learnt in the features.

- An alternative training strategy is proposed to achieve an unbiased classifier and retrain discriminative power of the feature representation. This training scheme is generic and can be adopted to other feature transfer learning algorithm. Various visualization results have shown that our approach can indeed increase the intra-class variation for long-tail classes.

1.3 Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 gives more background introduction and reviews related work on face recognition. Chapter 3 develops the proposed multi-task CNN framework for PIE-invariant face recognition. Chapter 4 explores large-pose face frontalization by using prior of 3D Morphable Models. Chapter 5 presents a feature transfer learning approach for face recognition with long-tail data. Chapter 6 concludes this dissertation.

Chapter 2

Background and Related Work

Biometrics refer to the technologies to recognize humans by their physical traits or behavior traits. Physical traits, including iris [32], fingerprint [94], face [135], etc., are believed to be more reliable than behavior traits such as voice [103], typing behavior [109], gait [147], and so on. Among these physical traits, face has the advantage of being non-invasive compared to iris and fingerprint. In the deep learning era, large amount of training data is of essential importance to the development of a recognition algorithm. Collecting face images online is a much easier task than collecting iris or fingerprint from human subjects. As a result, face recognition has advanced a lot with the development of deep learning techniques.

This section introduces basic knowledge that is necessary to understand deep face recognition and reviews related work.

2.1 Basics

2.1.1 Recognition Categories

Depending on the application scenarios, face recognition can be operated in two modes: face verification (or authentication), and face identification (or recognition) [65].

Face Verification Face verification is a one-to-one matching problem. Giving two face images, the problem is to answer whether they belong to the same identity or not. It involves comparing the

feature similarity with a pre-defined threshold to make the decision. Typical application scenarios include unlocking personal devices, or self-serviced check-in with ID.

Face Identification Face identification is a one-to-many matching problem. It includes a pre-registered gallery set with known identities and a query or probe image with unknown identity. If this probe image belongs to one of the gallery identities, it is a close-set recognition problem. The probe image is compared to each of the gallery images. The most similar face image is retrieved to identify the query image. However, in open-set face identification, a pre-defined threshold is needed to accept the query as one of the gallery identity or reject it as not present in this gallery set. Typical application scenarios include security door unlocking, or surveillance watch list searching.

In either face verification or identification, the common problem is to compare the similarity between two face representations. The discriminative power of the representation is the essential component to successful face recognition.

2.1.2 Face Recognition Pipeline

As discussed above, face recognition can be simplified as the problem of comparing the similarity between a pair of face images. The recognition pipeline is shown in Figure 2.1. It consists of four steps: face detection, face alignment, feature extraction, and face matching.

Face Detection Face detection is a necessary first-step in face recognition systems [58]. It is the process to detect the region of a face (blue box in Figure 2.1). State-of-the-art face detection algorithms [107, 67, 117] can usually achieve satisfactory pre-processing results on current face benchmarks. However, detecting faces with very low resolution is still challenging.

Face Alignment Face alignment is the process to detect facial landmarks located around eyebrows, eyes, nose, mouth, and facial contour. The landmarks are used to align a face image to a canonical view. Even though large-pose face alignment [68, 69, 70] is challenging, it is not a big concern for

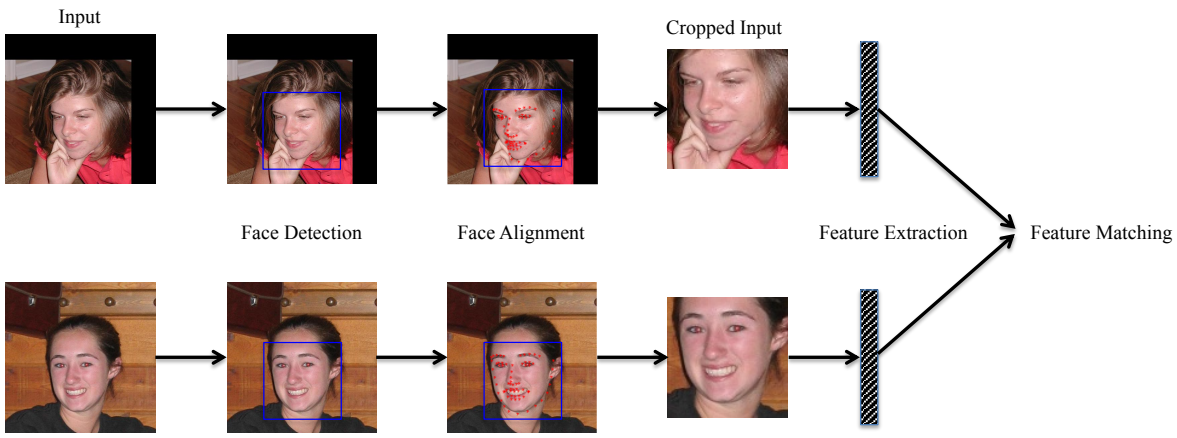


Figure 2.1: Face matching pipeline. It consists of four steps: face detection, face alignment, feature extraction, and feature matching.

deep face recognition as the face images are only required to be roughly aligned.

Feature Extraction Feature extraction is the most crucial step in the face matching pipeline. The history of face recognition can be viewed as the progress of identity feature learning, from the very early manual measurement [170], to hand-crafted features [34], and now to deep features [115]. Deep face recognition involves training a model from a large set of data and using this model as the feature extractor during the testing stage. Training such a model is the major component of a face recognition system, which is reviewed in Section 2.3.

Feature Matching Face matching is to compute the similarity, or distance between two face representations. Typical distance measurements are either Cosine distance between the original features or Euclidean distance between the L2-normalized features.

2.1.3 Evaluation Metrics

Face recognition performance is reported on three standard tasks including face verification, close-set and open-set face identification [65]. We will give a brief introduction on the evaluation metrics that are used in this dissertation.

Face Verification As discussed in Section 2.1.2, the face matching process calculates a distance between a pair of face images. This distance is compared to a pre-defined threshold. For distance that is smaller than this threshold, the face pair is considered as the same subject. Otherwise, it is considered as different subjects. Correctly verified pairs are named *true positive* (same-person pair) or *true negative* (different person pair). There are two types of errors: *false positive* (or *face acceptance*) refers to the mistake of recognizing different subjects as the same one; *false negative* (or *false rejection*) refers to the mistake of recognizing the same subject as different ones. Based on these two types of errors, face verification performance is evaluated using the following metrics:

- *Accuracy*: the percentage of truly recognized pairs (both *true positive* and *true negative*).
- *Equal Error Rate (EER)*: the error when *false positive rate (FPR)* and *false negative rate (FNR)* are the same, which is found by varying the threshold.
- *Receiver Operating Characteristic (ROC)*: the curve of *true positive rate (TPR)* against *false positive rate (FPR)* that are calculated by varying the threshold.
- *Area Under Curve (AUC)*: the percentage of area under the ROC curve.

Close-Set Face Identification Close-set face identification involves comparing the query face with each of the gallery face image. The resulting distances are sorted and ranked. The top n subjects with the closest distances are retrieved. A true match can be defined as when the true identity is observed in the top n ranks. *True Positive Identification Rate (TPIR)* refers to the probability of observing the true identity in the top n ranks. Evaluation metrics for close-set face identification based on TPIR are defined as follows:

- *Rank-n Accuracy*: TPIR at the rank of n. Typical values for n are 1, 5, 10.
- *Cumulative Match Characteristic (CMC)*: the curve of TPIR against ranks.

Open-Set Face Identification Open-set face identification is more complicated than the close-set identification. It adds an additional step to compare the smallest distance with a pre-defined threshold to determine whether the query exists in the gallery or not. This dissertation does not consider this scenario as most face benchmarks are prototyped for face verification and close-set face identification. The evaluation of open-set face identification can be referred to [65].

2.2 Challenges

Robust face recognition is a challenging problem. This challenge can be viewed from two perspectives: data variance and method design.

2.2.1 Data Variance

The fact that the intra-subject difference caused by pose, illumination, expression (collectively represented as "PIE"), age, and etc. can often surpasses the inter-subject difference can challenge state-of-the-art face recognition systems. Additional difficulties come from the various occlusions, low image quality, and so on.

Generic face recognition algorithms [115, 139, 33] aim to design novel loss functions that can handle the data variance implicitly. Other algorithms [132, 169, 153, 37, 29, 10] propose to explicitly handle one or multiple of the above challenges. Among these variations, pose has been studies the most for the following reasons. First, the self-occlusions caused by pose variation is difficult for conventional face recognition algorithms that only works well on frontal faces, which affects the progress of face recognition techniques. Second, human head pose is relatively easy to model with the help of 3D face models [12]. The estimated pose labels make supervised learning possible in the designing of pose-invariance face recognition algorithms.

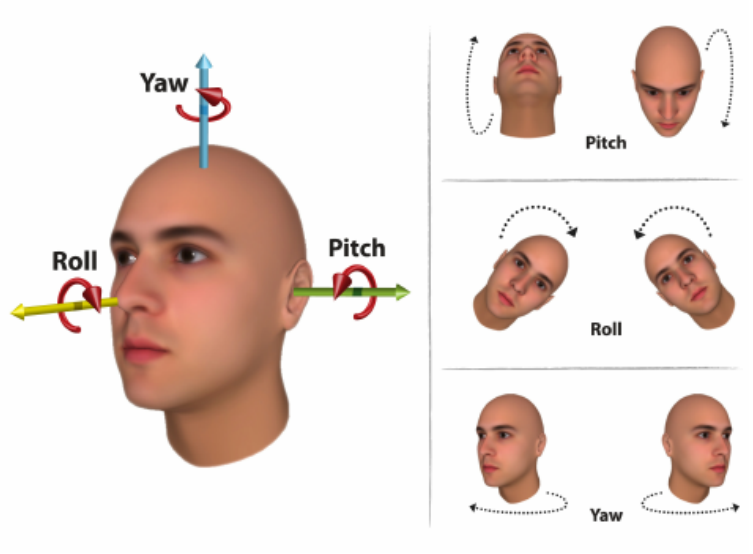


Figure 2.2: Pitch, yaw, roll angles for pose variation representation [4].

Pose Model As shown in Figure 2.2, human head has three degrees of freedom in rotation, which result in pitch, yaw, and roll angles for pose variation. Among these angles, the in-plane rotation caused by roll angle can easily be corrected in the face pre-processing step. Pitch angle is more widely observed in surveillance scenarios but not in the existing benchmark datasets like LFW [62], IJB-A [77], or MS-celeb-1M [49]. Yaw angle is the most studied because it is the most commonly observed. This dissertation mainly focuses on handling yaw angle in face recognition.

2.2.2 Method Design

Face recognition is a different problem compared to generic object recognition. First of all, many studies [73, 133, 80] in electrophysiology have uncovered evidence that visual cortex contain special regions involved in processing faces but no other objects, which shows that human brain separates face recognition from generic object recognition. From the machine learning perspective, face recognition is a fine-grained recognition problem. That is, the task is not to recognize whether there is a face or not, rather, it is to recognize the subject identity.

In generic object recognition, the training and testing stage usually involve recognizing the same classes, like the $1,000$-class ImageNet recognition challenge [112]. Typical method aims to learn a feature extractor and a classifier that perform well on the training set and then applies them to the testing set. In the scenario of zero-shot learning, i.e., recognizing novel objects without any training images available, other source of data like text are needed to learn the mapping from image to text [100, 121]. Since image and text are quite different from each other, zero-shot learning in generic object recognition is a very difficult task.

On the contrary, face recognition usually requires training on a set of subjects and test on unknown subjects. This is achievable without the need for any other domain information, which is different from zero-shot learning in generic object recognition. Because human face is well structured, it is possible to learn a mapping from the face image space to a feature space that can generalize to unseen subjects. How to learn such a mapping is the key in designing a face recognition algorithm.

Most deep neural network (DNN)-based methods treat face recognition as a traditional classification problem where the main focus is to design novel loss functions to guide feature learning. A good feature distribution is believed to have small intra-class variance and large inter-class variance. This has been a default rule in designing recognition methods. While generic object recognition algorithms can be applied to train a face recognition system, it is more desirable if any face prior information can be incorporated in the feature learning process. This is the challenging part when designing a face recognition method.

2.3 Methodologies

There are two major components in face recognition algorithms: feature extraction and loss design. Early non deep learning-based methods usually combine hand crafted features [2, 34, 91] with unsupervised or supervised subspace learning methods such as PCA [142], LDA [9] and Joint Bayesian [18]. With the development of deep neural networks, the above two-step procedure can be combined into an end-to-end learning framework. The main idea is to learn a mapping function from the input image space into a target space where a simple distance metric calculated on the features can approximate the semantic distance in the input space.

This section reviews related work in deep face recognition. First, we review methods for general face recognition where no other label except identity is used during training. The goal is to design a framework with advanced loss functions that can handle different challenges implicitly. This is in contrast to other face recognition algorithms where pose, expression, age information is used explicitly during training for robust face recognition w.r.t. specific variations. Among these variabilities, we focus on the review of pose-invariant face recognition algorithms.

2.3.1 General Deep Face Recognition

Based on how the loss function is designed, general deep face recognition algorithms can be classified into two categories [28]: probabilistic-based models and energy-based models. Given an input face image, probabilistic models aim to assign a normalized probability to each subject. Energy-based models (or metric learning) assign an un-normalized energy to a pair of face images. In the form of deep neural networks, probabilistic models learn a linear classifier on the features with cross-entropy loss. Energy-based models associate an energy to a pair of input faces by compare the feature similarities, and thus bypass the need for a linear classifier.

2.3.1.1 Probabilistic-based Models

Typical methods in this category consists of the softmax cross entropy loss and its variants. In the DNN framework, a fully connect layer is added after the features to output the logits, which is then converted to a probability distribution over all subjects with a softmax operation. Cross Entropy (CE) loss aims to maximize the probability of an input face belonging to the target label. For clarity, the softmax operation and the CE loss are together represented as softmax loss in the remaining of this dissertation. Softmax loss is widely used in different kinds of classification tasks due to its effectiveness and simplicity. However, it is also well known that the features learnt by softmax loss is not discriminative enough. Therefore, many methods aim to improve softmax loss from several different perspectives.

Feature and Weight Normalization Prior work have observed several problems with face recognition algorithms trained with softmax loss. First, it is proved in [138] that softmax loss can be minimized by increasing the feature norm. This can be easily achieved especially for good samples. In fact, Ranjan et al. [106] have observed that features with good quality are of higher norm than those with bad quality. Second, softmax loss uses inner product for classification during training while face matching is done by calculating the cosine distance or normalized Euclidean distance, such a gap will affect the performance as observed in [138]. Based on these observations, recent work explore the normalization of features or weights, or both when using softmax loss.

Ranjan et al. [106] propose a L_2-constrained softmax loss where the features are constrained with a norm of α. The bound of α is also discussed w.r.t. the number of classes and the probability score. Such a loss has proved to work better than the baseline softmax loss for face verification. Wang et al. [138] propose NormFace to reformulate the softmax loss where both the feature and the weight are normalized. Recently, ring loss is proposed by Zheng et al. [165] to regularize the

features to a target norm. This formulation converts the traditional normalization operation to a convex optimization problem. All the above studies have shown the effectiveness of performing normalization in softmax loss.

Auxiliary Losses The weight matrix in the final classification layer learns a template for each class. Softmax loss encourages the samples to be close (in the form of inner product) to its template while away from other classes' templates, which can guarantee the inter-class separation. However, it does not explicitly model the intra-class variation. Therefore, some work propose additional losses with an objective to increase the intra-class compactness.

The well-known center loss [141] proposes to learn a center for each class and penalize the distance between the sample features to their corresponding class centers. With the joint supervision of softmax loss and center loss, the discriminative power of the features is significantly improved. Motivated by center loss, Qi and Su propose contrastive-center loss [104] that considers both inter-class separation and intra-class compactness. Similarly, He et al. [57] propose triplet-center loss to use centers to formulated triplet for 3D object retrieval.

Margins in Softmax Loss Besides introducing normalization or adding auxiliary losses, another category of methods is to add margins in the softmax loss. Liu et al. [88] propose large-margin softmax loss (L-Softmax) with inter-class angular margin constraint that controls the learning difficulty. Later on, SphereFace [87] is proposed, which shares similar idea with L-Softmax loss but with the weight vectors being normalized. Now it is widely accepted that both the feature and the weight vector need to be normalized, and a scale layer is added after the feature normalization to make it an easier optimization task. Under such a framework, CosFace [139] and ArcFace [33] are proposed with margins added to the cosine value or the angle. However, one problem with these two loss functions are the tedious effort for parameter tuning of the scale and margin.

2.3.1.2 Energy-based Models

Different from probabilistic-based models which estimate a normalized probability distribution over all classes, energy-based models assign an unnormalized energy to an input pattern. The key differences are in three folds. First, maximizing the probability of one sample belonging to the target class will automatically minimize the probabilities of it belonging to other non-target classes. On contrary, energy-based models need to sample same-person pairs and different-person pairs to make contrastive loss functions that minimize intra-class distances and maximize inter-class distances. Second, one weight vector is required for each class in the probabilistic-based models, which is limited by the memory constraint when the number of classes is large. Energy-based models compare the energy between different input patterns and thus do not need to learn a weight vector for each class. Third, training one sample in probabilistic-based models is equal to make C (the number of classes) comparisons between the sample and all weight vectors, which is more efficient than energy-based models that makes a few limited comparisons at a time. Therefore, there are three major components in designing an energy-based model: contrastive term in the loss function, sample selection, and comparison scheme.

Contrastive loss [28] is the pioneer work of energy-based models for face recognition. It proposes to use a scalar energy function that is 0 for same-person pair and 1 for different-person pair. Ten years later, Schoroff et al. propose FaceNet [115] that generalizes the contrastive loss to triplet loss, which minimizes the distance between an anchor and a positive sample that belongs to the same identity and maximizes the distance between the anchor and a negative sample of a different identity. Triplet loss achieves large performance gain on the LFW database. However, it is hard to train in practice where the bottleneck is how to select meaningful triplet samples.

Other metric learning methods aim to generalize triplet loss by improving the sample selection

or comparison scheme. For example, Song et al. [99] propose lifted structured feature embedding that considers all same- and different-person pairs of comparisons in each mini-batch. Kihyuk [122] proposes N-pair loss that constructs a mini-batch with the comparisons of one positive pair and $N-1$ negative pairs. Zhang et al. [160] propose range loss that considers the hardest intra-class and inter-class distances in each mini-batch. Although range loss considers both inter-class and intra-class variations, it still has to combine with softmax loss to achieve desirable performance. Different from the above approaches that try to generalize triplet loss by exploring different sample selection methods, Ustinova and Lempitsky propose histogram loss [136] that focus on the comparison scheme. Histogram loss aims to separate two distance distributions of positive pairs and negative pairs. It achieves favorable results and has no tunable parameters.

The methods reviewed in this section is general and applicable to both generic objects and faces. Since it is not the focus of this dissertation, we use softmax loss if not specified particularly.

2.3.2 Pose-Invariant Face Recognition

Face image observed from extreme views is a problem of fundamental interest in both human and machine facial processing and recognition. Indeed, while humans are innately skilled at face recognition, newborns do not perform better than chance on recognition from profile views [134], although this ability seems to develop rapidly within a few months of birth [39]. Similarly, PIFR remains an enduring challenge in computer vision. As discussed in [36], current PIFR algorithms can be broadly grouped into three categories while some work combines the three techniques.

- *Multi-View Subspace Learning.*

  Methods in this category aim to project the face representation from different view-points to the same subspace where face matching is meaningful.

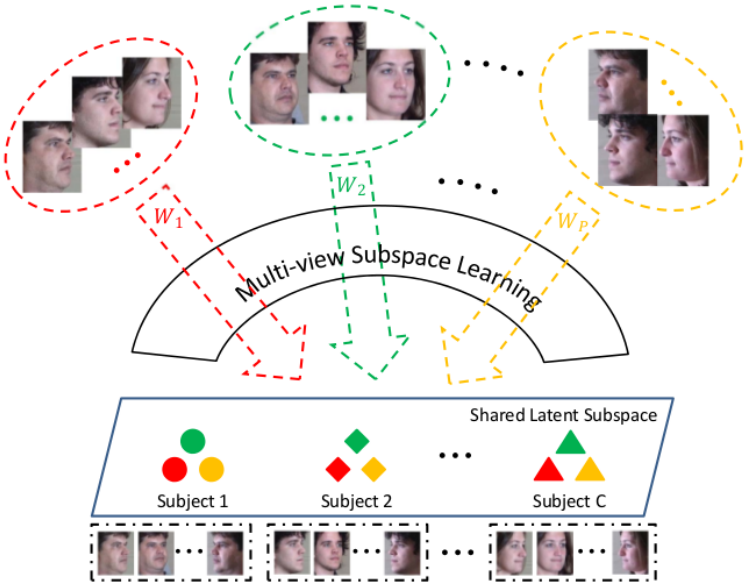


Figure 2.3: The general framework for multi-view subspace learning [36].

- *Pose-Invariant Feature Extraction.*

  Methods in this category usually rely on a large number of labeled dataset for pose-invariant feature extraction and conventional classifiers for face matching.

- *Face Synthesis.*

  Methods in this category propose to synthesize face images with a target pose from a source pose so that faces under different views can be compared at the same view point.

We will review some prior work in these three categories.

2.3.2.1 Multi-View Subspace Learning

The pose-variant face images are distributed in a high-dimensional non-linear manifold. As shown in Figure 2.3, multi-view subspace learning-based methods aim to learn the projections from different views to the same subspace where the comparison of face images with different poses make sense. Li et al. [81] propose Canonical Correlation Analysis (CCA) for the subspace learning

where two projection matrices are learnt with one for each pose. The goal of CCA is to maximize the correlation of the projected samples of the same subject with two different poses.

Rupnik and Shawe-Taylor [111] present Multi-View CCA (MCCA) that extend CCA to work for multiple pose variation in the training set instead of two different poses in the original CCA. CCA and MCCA are both linear models, which is limited because the texture change caused by pose is non-linear. A natural extension of CCA is to include kernel technique as proposed in [3]. The multi-view subspace learning-based methods all require multiple pose-variant images for each subject in the training set, which is hard to achieve for in-the-wild datasets.

2.3.2.2 Pose-Invariant Feature Extraction

The rotation of human head results in the loss of semantic correspondence between pose-variant face images. One solution in pose-invariant feature extraction is to rebuild the semantic correspondence. Specifically, previous work rely on the facial landmark detection in order to extract features around each keypoint. For example, Biswas et al. [11] propose to extract SIFT [92] features at each landmark and concatenate the features of each landmark as the pose-robust feature representation. Chen et al. [19] propose to extract a high dimensional LBP representation from the patches around the facial landmarks for PIFR.

Accurate landmark alignment is crucial for the above methods. However, face alignment in unconstrained images is still a challenging problem. Another solution for pose-invariant feature extraction is to build the semantic correspondence between two images based on the extracted keypoints instead of the fixed facial landmarks. For example, Liao et al. [84] develop a partial face recognition approach based on Multi-Keypoint Descriptors (MKD) that does not require face alignment. The probe face image is represented sparsely by a set of gallery descriptors.

Besides the engineering features, learning-based features become more popular recently. Neu-

ral networks are employed and shown to be better at handling large pose variations due to the high non-linear property and huge amount of training data. For example, Zhu et al. [168] build a deep neural network for Face Identity-Preserving (FIP) feature extraction. Zhang et al. [161] propose an auto-encoder for pose-robust feature extraction where the pose-variant inputs are learnt to map to a random face. The rationality behind this approach is that if all pose-variant inputs can map to the same random face, the learnt representation should be pose-invariant. Deep models are better at extracting pose-invariant features. It also requires a large amount of training data and lots of efforts for network configuration.

Recently there is an increasing interest in disentangled representation learning for PIFR. The pioneer work of DR-GAN [132] introduces an encoder-decoder structured GAN framework to learn disentangled representation via rotating face images. Peng et al. [102] propose a framework for feature disentanglement via side-task estimations and reconstruction. Zhao et al. [164] propose to a Pose Invariant Model (PIM) that combines a dual-path face frontalization branch and a discriminative learning branch. All the above methods have achieved good performance for PIFR as well as appealing visual results for image synthesis.

2.3.2.3 Face Synthesis

Face synthesis approaches aim to generate face images at a different view. This is motivated by the fact that comparing faces with the same view is an easier task than comparing face images observed at different views. The majority of the methods in this category is proposed for face frontalization, which is defined as the process of generating a frontal-view face image from an input image with arbitrary pose. This is a challenging problem because recovering the missing information caused by self-occlusion is ambiguous and ill-posed.

Early attempt for face frontalization is based on piece-wise affine warping for pose normaliza-

tion [41, 7]. Recent work on face frontalization rely on 3DMM for face rotation. For example, Zhu et al. [167] propose a High-fidelity Pose and Expression Normalization (HPEN) method to generate a natural face with frontal view and neutral expression. The pose and expression variations are eliminated in the estimated 3DMM that is achieved via 2D landmark fitting. An improved face recognition rate is observed for Multi-PIE and LFW. Similarly, Hassner et al. [40] explores unconstrained face frontalization by using a single 3D surface model for all query images. It is shown to be an efficient and effective alternative for the personalized 3DMM fitting.

Neural network-based methods are employed for face frontalization. The work of [151] proposes a deep neural network for face rotation. Given an input image, the proposed framework can generate a face image of the subject with the target pose specified by the remote code, which is concatenated to the image boundary. More recently, Disentangled-Representation learning Generative Adversarial Network (DR-GAN) is proposed for PIFR. It can generate face images with a target pose specified by the pose code. By employing the GAN [44] framework, DR-GAN is very effective in both representation learning and face rotation.

Chapter 3

Multi-Task Learning

3.1 Introduction

Multi-task learning (MTL) aims to learn several tasks *simultaneously* to boost the performance of the main task or all tasks. It has been successfully applied to face detection [20, 156], face alignment [162], pedestrian detection [131], attribute estimation [1], and so on. Despite the success of MTL in various vision problems, there is a lack of comprehensive study of MTL for face recognition. In this work, we study face recognition as a multi-task problem where identity classification is the main task with PIE estimations being the side tasks. We answer the questions of how and why PIE estimations can help face recognition.

We incorporate MTL into the CNN framework for face recognition. It is widely assumed in MTL that different tasks share the same features. Traditional linear models can be applied where each task is parameterized by a weight vector. The weight vectors of all tasks form a weight matrix $\mathbf{W}$, which is regularized by $l_{2,1}$ norm [5] or trace norm [66] to encourage $\mathbf{W}$ to be a low-rank matrix. In our work, the shared features are learnt through several convolution and pooling layers. A fully connected layer is added to the shared features for the classification of each task. We observe that the side tasks serve as regularizations to learn more discriminative and disentangled identity features for PIE-invariant face recognition.

As shown in Figure 3.1, when identity (x axis) is mixed with pose variation (y axis), single-task

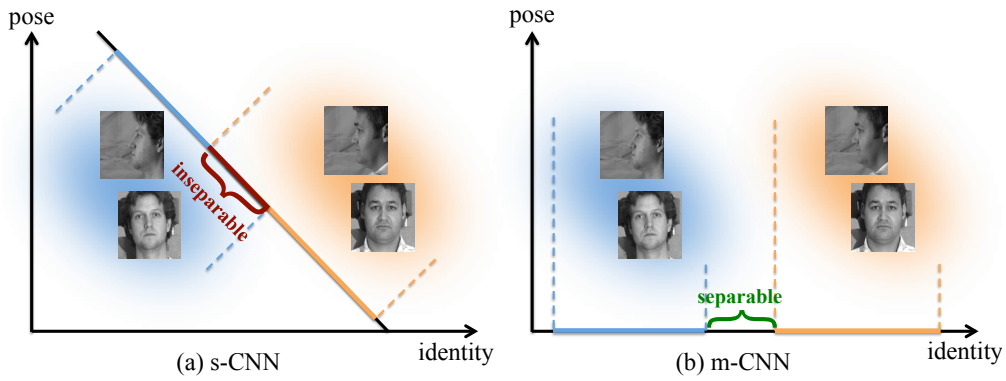


Figure 3.1: We propose MTL to disentangle the PIE variations from learnt identity features. (a) For single-task learning, the main variance is captured in x-y, resulting in an inseparable region between these two subjects. (b) For multi-task learning, identity is separable in x-axis by excluding y-axis that models pose variation.

learning (s-CNN) may learn a joint decision boundary along x-y, resulting in an inseparable region between the two different subjects. In contrary, with multi-task learning, the shared features are learnt to model identity and pose separately. The identity features can exclude pose variation by selecting only the key dimensions that are essential for face recognition, which leads to PIFR.

One problem in MTL is how to weight the importance of different tasks. Prior work either treat different tasks equally [151] or obtain the weights via greedy search [131]. However, it is time consuming or practically impossible to find the optimal weights for all side tasks via brute-force search in a CNN framework. To solve this problem, we propose dynamic weights where we only determine the overall importance for all side tasks, and the CNN learns to dynamically assign a loss weight to each side task during training, which is efficient and effective.

Since pose variation is the most challenging one among other non-identity variations, and the proposed multi-task CNN (m-CNN) already classifies all training images into different pose groups, we propose to apply divide-and-conquer to CNN learning. Specifically, we develop a novel pose-directed multi-task CNN (p-CNN) where the pose estimation can categorize the training data into three different pose groups (left, frontal, and right), direct them through different routes in the

network to learn pose-specific identity features in addition to the generic identity features. Similarly, the loss weights for extracting these two types of features are determined dynamically. In the testing stage, a stochastic routing scheme is formulated for feature matching, which is effective in handling pose variation in face recognition.

Multi-PIE [45] is an ideal dataset to study face recognition under PIE variations. It has been used to study face recognition robust to pose [82, 71], illumination [53, 52], and expression [29, 167]. Most prior work study the combined variations of pose and illumination [37, 157, 169] with the increase of pose variations from half-profile [157] to a full range [145]. This work utilizes *all* data in Multi-PIE, i.e., faces with the full range of PIE variations, as the main experimental dataset. To the best of our knowledge, there is no prior face recognition work that studies the full range of variations in Multi-PIE. Further, we apply our method to in-the-wild datasets where we only consider pose as the side task and the estimated labels serve as the ground truth labels for training.

In summary, we make the following contributions.

- We formulate face recognition as an MTL problem and explore how it works via an energy-based weight analysis.
- We propose a dynamic-weighting scheme to learn the loss weights for each side task automatically in the CNN.
- We develop a pose-directed multi-task CNN to learn pose-specific identity features and a stochastic routing scheme for feature fusion during the testing stage.
- We perform a comprehensive and the first face recognition study on the entire Multi-PIE. We achieve comparable or superior performance to state-of-the-art methods on Multi-PIE, LFW [62], CFP [116], and IJB-A [77].

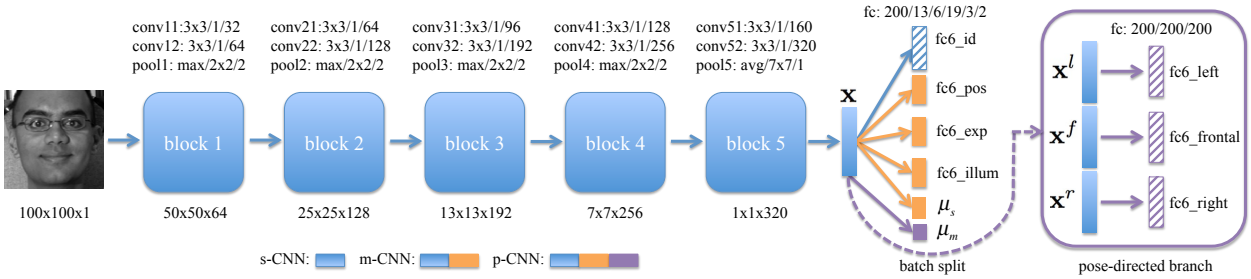


Figure 3.2: The proposed m-CNN and p-CNN for face recognition. Each block reduces the spatial dimensions and increases the channels of the feature maps. The formats for the convolution and pooling parameters are: filter size / stride / filter number and method / filter size / stride. The feature dimensions after each block operation are shown on the bottom. The dashed line represents the batch split operation as shown in Figure 3.3. The layers with the stripe pattern are the identity features used in the testing stage for face recognition.

3.2 Proposed Method

In this section, we demonstrate our methods on Multi-PIE dataset and extend it to unconstrained datasets in the experiments. First, we propose a multi-task CNN (m-CNN) with dynamic weights for face recognition (main task) and PIE estimations (side tasks). Second, we propose a pose-directed multi-task CNN (p-CNN) to tackle pose variation by separating all poses into different groups and jointly learning pose-specific identity features for each group.

3.2.1 Multi-Task CNN

We adapt CASIS-Net [150] as our backbone network with three modifications. First, batch normalization (BN) [64] is applied to accelerate training. Second, the contrastive loss is removed to simplify the loss function. Third, the dimension of the fully connected layer is changed according to different tasks. Details of the layer parameters are shown in Figure 3.2. The network consists of five blocks each including two convolution layers and a pooling layer. BN and ReLU [97] are used after each convolution layer, which are omitted for clarity. Similar to [150], no ReLU is used after conv52, and a dropout layer with a ratio of 0.4 is applied after pool5.

Given a set of N training images and their labels: $\mathbf{D} = \{\mathbf{I}_i, \mathbf{y}_i\}_{i=1}^{N}$, where each label $\mathbf{y}_i$ is a vector consisting of the identity label y_i^d and the side task labels. In this work, we consider three side tasks including pose (y_i^p), illumination (y_i^l), and expression (y_i^e). We eliminate the sample index i for clarity. As shown in Figure 3.2, the proposed m-CNN extracts a high-level feature representation $\mathbf{x} \in \mathbb{R}^{D\times 1}$:

$$\mathbf{x} = f(\mathbf{I}; \mathbf{k}, \mathbf{b}, \gamma, \beta), \tag{3.1}$$

where $f(\cdot)$ represents the non-linear mapping from the input image to the shared features. $\mathbf{k}$ and $\mathbf{b}$ are the sets of filters and bias of all convolution layers. γ and β are the sets of scales and shifts in all BN layers. Let $\Theta = \{\mathbf{k}, \mathbf{b}, \gamma, \beta\}$ denote all parameters to be learnt to extract the features.

The extracted features $\mathbf{x}$, which is pool5 in our model, are shared among all tasks. Suppose $\mathbf{W}^d \in \mathbb{R}^{D\times D_d}$ and $\mathbf{b}^d \in \mathbb{R}^{D_d\times 1}$ are the weight matrix and bias vector in the fully connected layer for identity classification, where D_d is the number of different identities in $\mathbf{D}$. The generalized linear model is applied:

$$\mathbf{y}^d = \mathbf{W}^{d\mathsf{T}}\mathbf{x} + \mathbf{b}^d. \tag{3.2}$$

$\mathbf{y}^d$ is then fed to a softmax layer to compute the probability of $\mathbf{x}$ belonging to each subject in the training set.

$$softmax(\mathbf{y}^d)_n = p(\hat{y}^d = n|\mathbf{x}) = \frac{\exp(\mathbf{y}_n^d)}{\sum_j \exp(\mathbf{y}_j^d)}, \tag{3.3}$$

where $\mathbf{y}_j^d$ is the jth element in $\mathbf{y}^d$. The $softmax(\cdot)$ function converts the model output $\mathbf{y}^d$ to a probability distribution over all subjects and the subscript selects the nth element. Finally, the estimated identity $\hat{y}^d$ is obtained via:

$$\hat{y}^d = \underset{n}{\text{argmax}} \quad softmax(\mathbf{y}^d)_n. \tag{3.4}$$

Then the cross-entropy loss can be employed:

$$L(\mathbf{I}, y^d) = -\log(p(\hat{y}^d = y^d | \mathbf{I}, \Theta, \mathbf{W}^d, \mathbf{b}^d)). \tag{3.5}$$

Similarly, we formulate the losses of the side tasks. Let $\mathbf{W} = \{\mathbf{W}^d, \mathbf{W}^p, \mathbf{W}^l, \mathbf{W}^e\}$ represent the weight matrices for identity and PIE classifications. The bias terms are eliminated for simplicity. Given the training set $\mathbf{D}$, our m-CNN aims to minimize the combined loss of all tasks:

$$\underset{\Theta, \mathbf{W}}{\operatorname{argmin}} \quad \alpha_d \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \alpha_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \alpha_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \alpha_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e), \tag{3.6}$$

where α_d, α_p, α_l, α_e control the importance of each task. It becomes a single-task model (s-CNN) when $\alpha_p = \alpha_l = \alpha_e = 0$. The loss drives the model to learn both the parameters Θ for extracting the shared features and $\mathbf{W}$ for the classification tasks. In the testing stage, the features before the softmax layer ($\mathbf{y}^d$) are used for face recognition by applying a face matching procedure based on cosine similarity.

3.2.2 Dynamic-Weighting Scheme

In CNN-based MTL, it is an open question on how to set the loss weight for each task. Prior work either treat all tasks equally [151] or obtain the weights via brute-force search [131]. However, it is very time-consuming to search for all combinations. To solve this problem, we propose a dynamic-weighting scheme to automatically assign loss weights to each side task during training.

First, the weight for the main task is set to 1, i.e. $\alpha_d = 1$. Second, instead of finding the loss weight for each task, we find the summed loss weight for all side tasks, i.e. $\varphi_s = \alpha_p + \alpha_l + \alpha_e$, via brute-force search in a validation set. Our m-CNN learns to allocate φ_s to the side tasks. As shown

in Figure 3.2, we add a fully connected layer and a softmax layer to the shared features $\mathbf{x}$ to learn the dynamic weights. Let $\omega_s \in \mathbb{R}^{D\times 3}$ and $\varepsilon_s \in \mathbb{R}^{3\times 1}$ denote the weight matrix and bias vector in the new added fully connected layer,

$$\mu_s = softmax(\omega_s{}^{\intercal}\mathbf{x} + \varepsilon_s), \tag{3.7}$$

where $\mu_s = [\mu_p, \mu_l, \mu_e]^{\intercal}$ are the dynamic loss weights for the side tasks with $\mu_p + \mu_l + \mu_e = 1$. So (3.6) becomes:

$$\begin{aligned} \underset{\Theta,\mathbf{W},\omega_s}{\operatorname{argmin}} \quad & \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \varphi_s \Big[\mu_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \mu_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \mu_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e) \Big] \\ & s.t. \quad \mu_p + \mu_l + \mu_e = 1. \end{aligned} \tag{3.8}$$

We use mini-batch stochastic gradient descent to solve the above optimization problem where the dynamic weights are averaged over a batch of samples. Intuitively, we expect our m-CNN to behave in two different aspects in order to minimize the loss. First, since our main task contribute mostly to the final loss (usually $\varphi_s < 1$), the side task with the largest contribution to the main task should have the highest weight in order to reduce the loss of the main task. Second, our m-CNN should assign a higher weight for an easier task with a lower loss so as to reduce the overall loss. We have observed these effects as shown in Figure 3.4 (a).

3.2.3 Pose-Directed Multi-Task CNN

Given the diverse variations in the data, it is very challenging to learn a non-linear mapping to estimate the correct identity from a face image with arbitrary PIE. This challenge has been encountered in classic pattern recognition work. For example, in order to handle pose variation, [83]

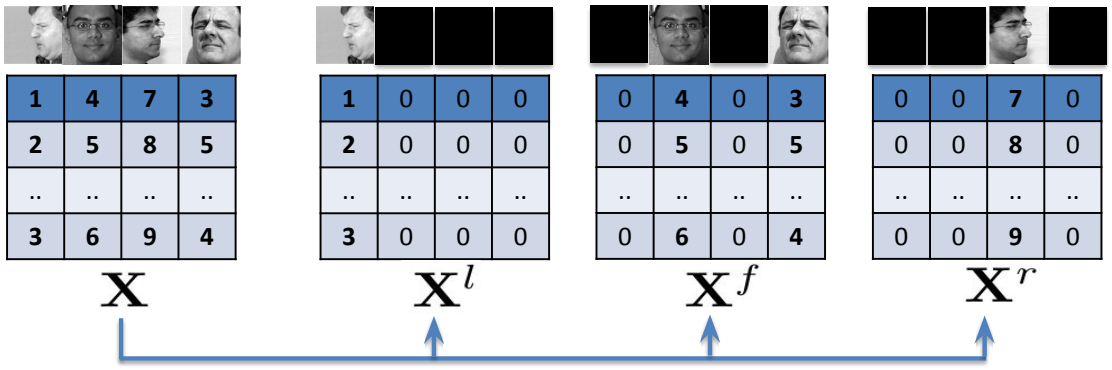


Figure 3.3: Illustration of the batch split operation in p-CNN.

proposes to construct several face detectors where each of them is in charge of one specific view. Such a divide-and-conquer scheme can be applied to CNN learning because the side tasks can "divide" the data and allow the CNN to better "conquer" them by learning tailored mapping functions. Therefore, we propose a novel task-directed multi-task CNN where the side task categorizes the training data into multiple groups and directs them to different routes in the network. Since pose is considered as the primary challenge in face recognition [145, 157, 169], we propose pose-directed multi-task CNN (p-CNN) to handle pose variation. However, it is applicable to any other variation.

As shown in Figure 3.2, p-CNN is built on top of m-CNN by adding the pose-directed branch (PDB). The PDB groups face images with similar poses to learn pose-specific identity features via a batch split operation. We separate poses into three groups: left profile (G^l), frontal (G^f), and right profile (G^r). As shown in Figure 3.3, the goal of batch split is to separate a batch of N_0 samples ($\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N_0}$) into three batches $\mathbf{X}^l$, $\mathbf{X}^f$, and $\mathbf{X}^r$, which are of the same size as $\mathbf{X}$. During training, the ground truth pose is used to assign a face image into the correct group. Let us take the frontal group as an example:

$$\mathbf{X}_i^f = \begin{cases} \mathbf{x}_i, & \text{if } y_i^p \in G^f \\ \mathbf{0}, & \text{otherwise,} \end{cases} \tag{3.9}$$

where $\mathbf{0}$ denotes a vector of all zeros with the same dimension as $\mathbf{x}_i$. The assignment of $\mathbf{0}$ is to guarantee valid input to the next layer when no sample is passed into one group. Therefore, $\mathbf{X}$ is separated into three batches where each batch consists of only the samples belonging to the corresponding pose group. Each group learns a pose-specific mapping to a joint space, resulting in three different sets of weights: $\{\mathbf{W}^l, \mathbf{W}^f, \mathbf{W}^r\}$. Finally, the features from all groups are merged as the input to a softmax layer to perform robust identity classification jointly.

Our p-CNN aims to learn two types of identity features: $\mathbf{W}^d$ are the weights to extract generic identity features that are robust to all poses; $\mathbf{W}^{l,f,r}$ are the weights to extract pose-specific identity features that are robust within a small pose range. Both tasks are considered as our main tasks. Similar to the dynamic-weighting scheme in m-CNN, we use dynamic weights to combine our main tasks as well. The summed loss weight for these two tasks is $\varphi_m = \alpha_d + \alpha_g$. Let $\omega_m \in \mathbb{R}^{D\times 2}$ and $\varepsilon_m \in \mathbb{R}^{2\times 1}$ denote the weight matrix and bias vector for learning the dynamic weights,

$$\mu_m = softmax(\omega_m{}^{\mathsf{T}}\mathbf{x} + \varepsilon_m). \tag{3.10}$$

We have $\mu_m = [\mu_d, \mu_g]^{\mathsf{T}}$ as the dynamic weights for generic identity classification and pose-specific identity classification. Finally, the loss of p-CNN is formulated as:

$$\begin{aligned} \underset{\Theta,\mathbf{W},\omega}{\operatorname{argmin}} \quad & \varphi_m \Big[\mu_d \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^d) + \mu_g \sum_{g=1}^{G}\sum_{i=1}^{N_g} L(\mathbf{I}_i, y_i^d)\Big] + \\ & \varphi_s \Big[\mu_p \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^p) + \mu_l \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^l) + \mu_e \sum_{i=1}^{N} L(\mathbf{I}_i, y_i^e)\Big] \\ & s.t. \quad \mu_d + \mu_g = 1, \quad \mu_p + \mu_l + \mu_e = 1, \end{aligned} \tag{3.11}$$

where G is the number of pose groups, and N_g is the number of training images in the g-th group. $\omega = \{\omega_m, \omega_s\}$ is the set of parameters to learn the dynamic weights for both the main and side

tasks. We set $\varphi_m = 1$.

Stochastic Routing During testing, we can use the estimated pose to direct an image to extract the pose-specific features. However, the pose estimation error may cause inferior feature extraction results, especially for unconstrained faces. To solve this problem, we propose a stochastic routing scheme. Specifically, given a test image $\mathbf{I}$, we extract the generic features ($\mathbf{y}^d$) and the pose-specific features ($\{\mathbf{y}^g\}_{g=1}^G$) by directing it to all paths in the PDB. We can also obtain the probabilities of the input image $\mathbf{I}$ belonging to each pose group as ($\{p^g\}_{g=1}^G$) using our pose estimation side task. The distance between a pair of face images ($\mathbf{I}_1$ and $\mathbf{I}_2$) is computed as:

$$s = \frac{1}{2} dist(\mathbf{y}_1^d, \mathbf{y}_2^d) + \frac{1}{2} \sum_{i=1}^{G} \sum_{j=1}^{G} dist(\mathbf{y}_1^i, \mathbf{y}_2^j) \cdot p_1^i \cdot p_2^j, \tag{3.12}$$

where $dist(\cdot)$ is the cosine distance between two feature vectors. The proposed stochastic routing accounts for all combinations of pose-specific features weighted by the probabilities. This is more robust to pose estimation errors. We treat the generic features and pose-specific features equally and fuse them for face recognition.

3.3 Experimental Results

We evaluate the proposed m-CNN and p-CNN under two settings: (1) face identification on Multi-PIE with PIE estimations being the side tasks; (2) face verification/identification on in-the-wild datasets including LFW, CFP, and IJB-A, where pose estimation is the only side task. Further, we analyze the effect of MTL on Multi-PIE and discover that the side tasks regularize the network to learn a disentangled identity representation for PIE-invariant face recognition.

Table 3.1: Comparison of the experimental settings that are commonly used in prior work on Multi-PIE. (* The 20 images consist of 2 duplicates of non-flash images and 18 flash images. In total there are 19 different illuminations.)

setting	session	pose	illum	exp	train sub. / images	gallery / probe	total	references
I	4	7	1	1	200 / 5,383	137 / 2,600	8,120	[6, 82]
II	1	7	20	1	100 / 14,000	149 / 20,711	34,860	[168, 151]
III	1	15	20	1	150 / 45,000	99 / 29,601	74,700	[145]
IV	4	9	20	1	200 / 138,420	137 / 70,243	208,800	[132, 169]
V	4	13	20	1	200 / 199,940	137 / 101,523	301,600	[154]
ours	4	15	20*	6	200 / 498,900	137 / 255,163	754,200	

3.3.1 Face Identification on Multi-PIE

Experimental Settings Multi-PIE dataset consists of 754,200 images of 337 subjects recorded in 4 sessions. Each subject was recorded with 15 different cameras where 13 at the head height spaced at 15° interval and 2 above the head to simulate a surveillance camera view, labeled as $\pm 45°$ in our work. For each camera, a subject was imaged under 19 different illuminations. In each session, a subject was captured with 2 or 3 expressions, resulting in 6 different expressions across all sessions. Unlike previous work that uses a subset of Multi-PIE for experiments, we use the entire dataset in our work (as shown in Table 3.1). The first 200 subjects are used for training. The remaining 137 subjects are used for testing, where one image with frontal pose, neutral illumination and neutral expression for each subject is selected as the gallery set and the remaining images are selected as the probe set.

We use the landmark annotations provided in [38] to align each face to a canonical view of size 100×100 with gray-scale. Similar to [141], we normalize the image by subtracting 127.5 and dividing by 128. We set momentum to 0.9 and weight decay to 0.0005. All models are trained for 20 epochs from scratch with a batch size of 4. The learning rate starts at 0.01 and reduces at 10th, 15th, and 19th epochs with a factor of 0.1. The output before the softmax layer is used as features for face matching based on cosine similarity. The rank-1 identification rate is reported for

face recognition. For the side tasks, the mean accuracy over all classes is reported.

We randomly select 20 subjects from the training set to form a validation set to find the optimal overall loss weight for all side tasks. We obtain $\varphi_s = 0.1$ via brute-force search. For p-CNN model training, we split the training set into three groups based on the yaw angle of the image: right profile ($-90°, -75°, -60°, -45°$), frontal ($-30°, -15°, 0°, 15°, 30°$), and left profile ($45°$, $60°, 75°, 90°$).

Effects of MTL Table 3.2 shows the performance comparison of single-task learning (s-CNN), multi-task learning (m-CNN), and pose-directed multi-task learning (p-CNN) on the entire Multi-PIE. First, we train four single-task models for identity (id), pose (pos), illumination (illum), and expression (exp) classification respectively. As shown in Table 3.2, the rank-1 identification rate of s-CNN is only 75.67%. The performance of the frontal pose group is much higher than those of the profile pose groups, indicating that pose variation is indeed a big challenge for face recognition. Among all side tasks, pose estimation is the easiest task, followed by illumination, and expression as the most difficult one. This is caused by two potential reasons: 1) discriminating expression is more challenging due to the non-rigid face deformation; 2) the data distribution over different expressions is unbalanced with insufficient training data for some expressions.

Second, we train multiple m-CNN models by adding only one side task at a time in order to evaluate the influence of each side task. We use "id+pos", "id+illum", and "id+exp" to represent these variants and compare them to the performance of adding all side tasks denoted as "id+all". To evaluate the effects of the dynamic-weighting scheme, we train a model with fixed loss weights for the side tasks as: $\alpha_p = \alpha_l = \alpha_e = \varphi_s/3 = 0.033$. The summation of the loss weights for all side tasks are equal to φ_s for all m-CNN variants in Table 3.2 for a fair comparison.

Comparing the rank-1 identification rates of s-CNN and m-CNNs, it is obvious that adding the side tasks is always helpful for the main task. The improvement of face recognition is mostly on the

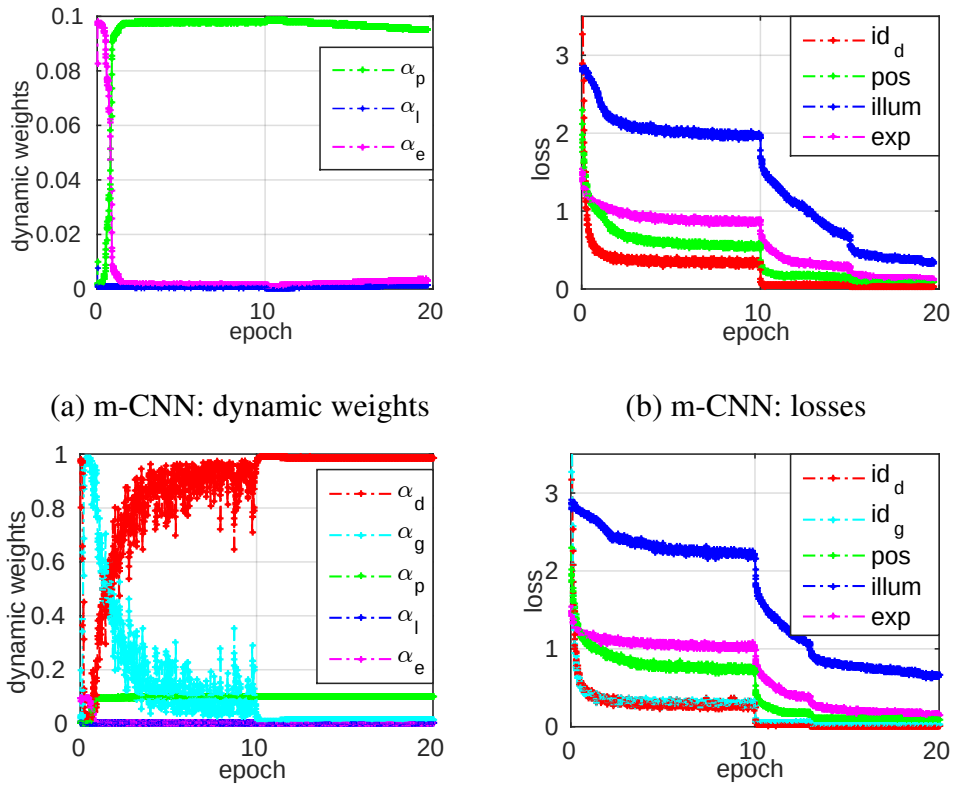
Table 3.2: Performance comparison (%) of single-task learning (s-CNN), multi-task learning (m-CNN) with its variants, and pose-directed multi-task learning (p-CNN) on the entire Multi-PIE dataset.

model	loss weights	rank-1 (all / left / frontal /right)	pose	illum	exp
s-CNN: id	$\alpha_d = 1$	75.67 / 71.51 / 82.21 / 73.29	–	–	–
s-CNN: pos	$\alpha_p = 1$	–	99.87	–	–
s-CNN: exp	$\alpha_l = 1$	–	–	96.43	–
s-CNN: illum	$\alpha_e = 1$	–	–	–	92.44
s-CNN: id+L2	$\alpha_d = 1$	76.43 / 73.31 / 81.98 / 73.99	–	–	–
m-CNN: id+pos	$\alpha_d = 1, \alpha_p = 0.1$	78.06 / 75.06 / 82.91 / 76.21	99.78	–	–
m-CNN: id+illum	$\alpha_d = 1, \alpha_l = 0.1$	77.30 / 74.87 / 82.83 / 74.21	–	93.57	–
m-CNN: id+exp	$\alpha_d = 1, \alpha_e = 0.1$	77.76 / 75.48 / 82.32 / 75.48	–	–	90.93
m-CNN: id+all	$\alpha_d = 1, \alpha_{p,l,e} = 0.033$	77.59 / 74.75 / 82.99 / 75.04	99.75	88.46	79.97
m-CNN: id+all (dynamic)	$\alpha_d = 1, \varphi_s = 0.1$	79.35 / 76.60 / 84.65 / 76.82	99.81	93.40	91.47
p-CNN	$\varphi_m = 1, \varphi_s = 0.1$	79.55 / 76.14 / 84.87 / 77.65	99.80	90.58	90.02

profile faces with MTL. The m-CNN "id+all" with dynamic weights shows superior performance to others not only in rank-1 identification rate, but also in the side task estimations. Further, the lower rank-1 identification rate of "id+all" w.r.t "id+pos" indicates that more side tasks do not necessarily lead to better performance without properly setting the loss weights. In contrast, the proposed dynamic-weighting scheme effectively improves the performance to 79.35% from the fixed weighting of 77.59%. As will be shown in Section 3.3.2, the side tasks in m-CNN help to inject PIE variations into the shared representation, similar to a regularization term. For example, an L2 regularization will encourage small weights. We add L2 regularization on the shared representation to s-CNN ("id+L2"), which improves over s-CNN without regularization. However, it is still much worse than the proposed m-CNN.

Third, we train p-CNN by adding the PDB to m-CNN "id+all" with dynamic weights. The loss weights are $\varphi_m = 1$ for the main tasks and $\varphi_s = 0.1$ for the side tasks. The proposed dynamic-weighting scheme allocates the loss weights to both two main tasks and three side tasks. P-CNN further improves the rank-1 identification rate to 79.55%.

Dynamic-Weighting Scheme Figure 3.4 shows the dynamic weights and losses during training



(a) m-CNN: dynamic weights

(b) m-CNN: losses

(c) p-CNN: dynamic weights

(d) p-CNN: losses

Figure 3.4: Dynamic weights and losses for m-CNN and p-CNN during the training process.

for m-CNN and p-CNN. For m-CNN, the expression classification task has the largest weight in the first epoch because it has the highest chance to be correct with random guess with the least number of classes. As training goes on, pose classification takes over because it is the easiest task (highest accuracy in s-CNN) and also the most helpful for face recognition (compare "id+pos" to "id+exp" and "id+illum"). α_p starts to decrease at the 11th epoch when pose classification is saturated. The increased α_l and α_e lead to a reduction in the losses of expression and illumination classifications. As we expected, the dynamic-weighting scheme assigns a higher loss weight for the easiest and/or the most helpful side task.

For p-CNN, the loss weights and losses for the side tasks behave similarly to those of m-CNN. For the two main tasks, the dynamic-weighting scheme assigns a higher loss weight to the easier

task at the moment. At the beginning, learning the pose-specific identity features is an easier task than learning the generic identity features. Therefore, the loss weight α_g is higher than α_d. As training goes on, α_d increases as it has a lower loss. Their losses reduce in a similar way, i.e., the error reduction in one task will also contribute to the other.

Compare to Other Methods As shown in Table 3.1, no prior work uses the entire Multi-PIE for face recognition. To compare with state of the art, we choose to use setting III and V to evaluate our method since these are the most challenging settings with more pose variation. The network structures and parameter settings are kept the same as those of the full set except that the outputs of the last fully connected layers are changed according to the number of classes for each task. Only pose and illumination are used as the side tasks.

The performance on setting III is shown in Table 3.3. Our s-CNN already outperforms c-CNN forest [145], which is an ensemble of three c-CNN models. This is attributed to the deep structure of CASIA-Net [150]. Moreover, m-CNN and p-CNN further outperform s-CNN with significant margins, especially for non-frontal faces. We want to stress the improvement margin between our method 91.27% and the prior work of 76.89% — a relative error reduction of 62%.

The performance on setting V is shown in Table 3.4. For fair comparison with FF-GAN [154], where the models are finetuned from pre-trained in-the-wild models, we also finetune s-CNN, m-CNN, p-CNN models from the pre-trained models on CASIA-Webface for 10 epochs. Our performance is much better than previous work with a relative error reduction of 60%, especially on large-pose faces. The performance gap between Table 3.3 / 3.4 and 3.2 indicates the challenge of face recognition under various expressions, which is less studied than pose and illumination variations on Multi-PIE.

Table 3.3: Multi-PIE performance comparison on setting III of Table 3.1.

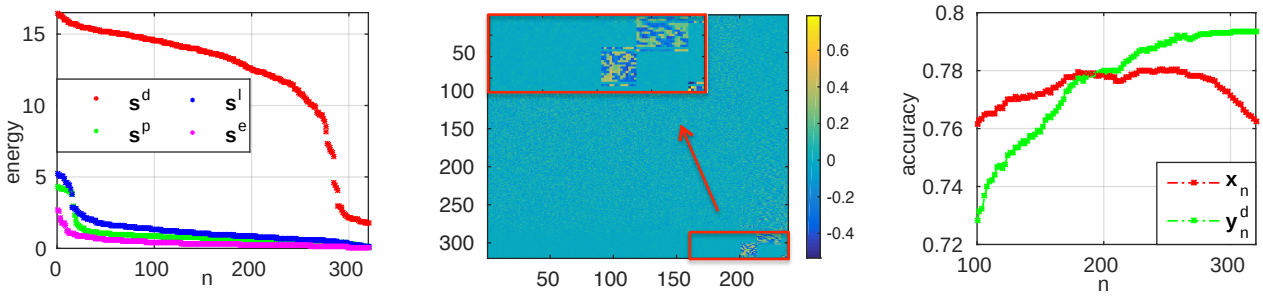
	$\pm 15^\circ$	$\pm 30^\circ$	$\pm 45^\circ$	$\pm 60^\circ$	$\pm 75^\circ$	$\pm 90^\circ$	avg.
Fisher Vector [118]	93.30	87.21	80.33	68.71	45.51	24.53	66.60
FIP_20 [168]	95.88	89.23	78.89	61.64	47.32	34.13	67.87
FIP_40 [168]	96.30	92.98	85.54	69.75	49.10	31.37	70.90
c-CNN [145]	95.64	92.66	85.09	70.49	55.64	41.71	73.54
c-CNN Forest [145]	96.97	94.05	89.02	74.38	60.66	47.26	76.89
s-CNN (ours)	98.41	96.89	85.18	88.71	82.80	76.72	88.45
m-CNN (ours)	99.02	97.40	89.15	89.75	84.97	76.72	90.08
p-CNN (ours)	**99.19**	**98.01**	**90.34**	**92.07**	**87.83**	**76.96**	**91.27**

Table 3.4: Multi-PIE performance comparison on setting V of Table 3.1.

	0°	$\pm 15^\circ$	$\pm 30^\circ$	$\pm 45^\circ$	$\pm 60^\circ$	$\pm 75^\circ$	$\pm 90^\circ$	avg.$[-60^\circ, 60^\circ]$	avg.$[-90^\circ, 90^\circ]$
FIP [168]	94.3	90.7	80.7	64.1	45.9	–	–	72.9	–
Zhu et al. [169]	95.7	92.8	83.7	72.9	60.1	–	–	79.3	–
Yim et al. [151]	**99.5**	95.0	88.5	79.9	61.9	–	–	83.3	–
DR-GAN [132]	97.0	94.0	90.1	86.2	83.2	–	–	89.2	–
FF-GAN [154]	95.7	94.6	92.5	89.7	85.2	77.2	61.2	91.6	85.2
s-CNN (ours)	95.9	95.1	92.8	91.6	88.9	84.9	78.6	92.5	89.2
m-CNN (ours)	95.4	94.5	92.6	91.8	88.4	85.3	82.2	92.2	89.6
p-CNN (ours)	95.4	**95.2**	**94.3**	**93.0**	**90.3**	**87.5**	**83.9**	**93.5**	**91.1**

3.3.2 How does m-CNN work?

It is well known in both the computer vision and the machine learning communities that learning multiple tasks together allows each task to leverage each other and improves the generalization ability of the model. For CNN-based MTL, previous work [163] has found that CNN learns shared features for facial landmark localization and attribute classifications, e.g. smiling. This is understandable because the smiling attribute is related to landmark localization as it involves the change of the mouth region. However, in our case, it is not obvious how the PIE estimations can share features with the main task. On the contrary, it is more desirable if the learnt identity features are disentangled from the PIE variations. Indeed, as we will show later, the PIE estimations regularize the CNN to learn PIE-invariant identity features.



(a) (b) (c)

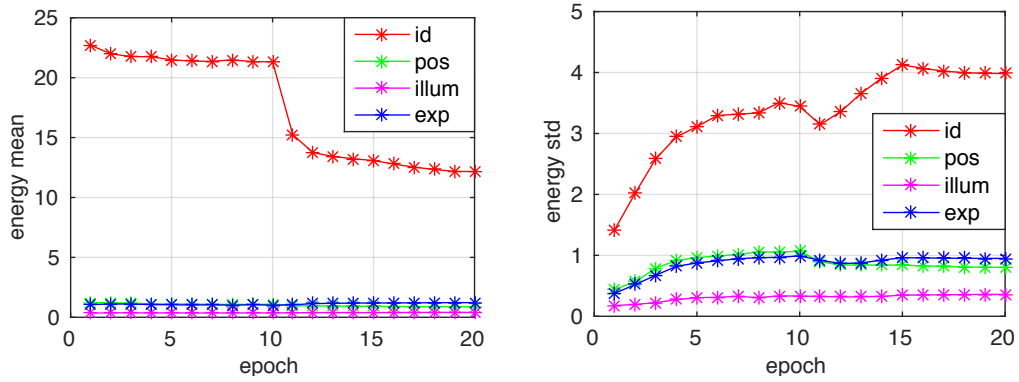
Figure 3.5: Analysis on the effects of MTL: (a) the sorted energy vectors for all tasks; (b) visualization of the weight matrix $\mathbf{W}^{all}$ where the red box in the top-left is a zoom-in view of the bottom-right content; (c) the face recognition performance with varying feature dimensions.

We investigate why PIE estimations are helpful for face recognition. The analysis is done on m-CNN model ("id+all" with dynamic weights) in Table 3.2. Recall that m-CNN learns a shared embedding $\mathbf{x} \in \mathbb{R}^{320\times 1}$. Four fully connected layers with weight matrices $\mathbf{W}^{d}_{320\times 200}$, $\mathbf{W}^{p}_{320\times 13}$, $\mathbf{W}^{l}_{320\times 19}$, $\mathbf{W}^{e}_{320\times 6}$ are connected to $\mathbf{x}$ to perform classification of each task (200 subjects, 13 poses, 19 illuminations, and 6 expressions). We analyze the importance of each dimension in $\mathbf{x}$ to each task. Taking the main task as an example, we calculate an energy vector $\mathbf{s}^{d} \in \mathbb{R}^{320\times 1}$ whose element is computed as:

$$\mathbf{s}^{d}_{i} = \sum_{j=1}^{200} | \mathbf{W}^{d}_{ij} | . \tag{3.13}$$

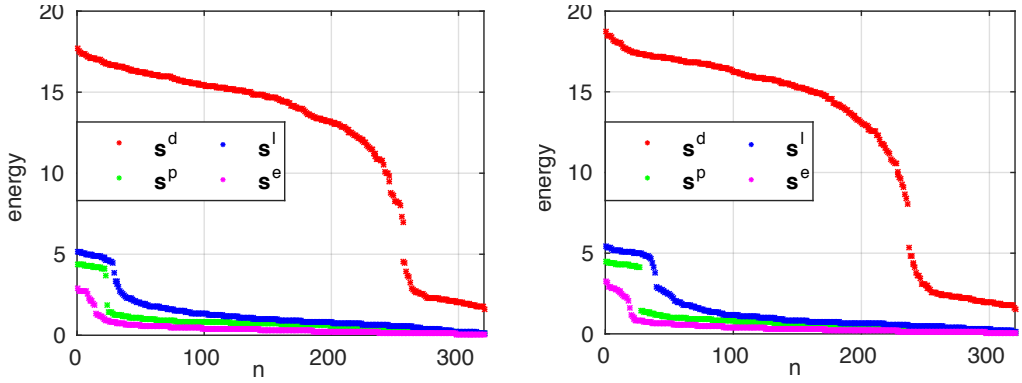
A higher value of $\mathbf{s}^{d}_{i}$ indicates that the ith feature in $\mathbf{x}$ is more important to the identity classification task. The energy vectors $\mathbf{s}^{p}$, $\mathbf{s}^{l}$, $\mathbf{s}^{e}$ for all side tasks are computed similarly. Each energy vector is sorted and shown in Figure 3.5 (a). For each curve, we observe that the energy distributes unevenly among all feature dimensions in $\mathbf{x}$. Note that the indexes of the feature dimension do not correspond among them since each energy vector is sorted independently.

To compare how each feature in $\mathbf{x}$ contributes to different tasks, we concatenate the weight matrix of all tasks as $\mathbf{W}^{all}_{320\times 238} = [\mathbf{W}^{d}, \mathbf{W}^{p}, \mathbf{W}^{l}, \mathbf{W}^{e}]$ and compute its energy vector as $\mathbf{s}^{all}$. We sort the rows in $\mathbf{W}^{all}$ based on the descending order in energy and visualize the sorted $\mathbf{W}^{all}$ in Figure 3.5



(a) mean (b) std.

Figure 3.6: The mean and standard deviation of each energy vector during the training process.



(a) $\varphi_s = 0.2$ (b) $\varphi_s = 0.3$

Figure 3.7: Energy vectors of m-CNN models with different overall loss weights.

(b). The first 200 columns represent the sorted $\mathbf{W}^d$ where most energy is distributed in the first $\sim$ 280 feature dimensions (rows), which are more crucial for face recognition and less important for PIE classifications. We observe that $\mathbf{x}$ are learnt to allocate a separate set of dimensions/features for each task, as shown in the block-wise effect in the zoom-in view. Each block shows the most essential features with high energy for PIE classifications respectively.

Based on the above observation, we conclude that the PIE classification side tasks help to inject PIE variations into the shared features $\mathbf{x}$. The weight matrix in the fully connected layer learns to select identity features and ignore the PIE features for PIE-invariant face recognition. To validate this observation quantitatively, we compare two types of features for face recognition:

1) $\mathbf{x}_n$: a subset of $\mathbf{x}$ with n largest energies in $\mathbf{s}^d$, which are more crucial in modeling identity variation; 2) $\mathbf{y}^d_{200\times 1} = {\mathbf{W}^d_{n\times 200}}^{\mathsf{T}}\mathbf{x}_{n\times 1} + \mathbf{b}^d$, which is the multiplication of the corresponding subset of $\mathbf{W}^d$ and $\mathbf{x}_n$. We vary n from 100 to 320 and compute the rank-1 face identification rate on the entire Multi-PIE testing set. The performance is shown in Figure 3.5 (c). When $\mathbf{x}_n$ is used, the performance improves with increasing dimensions and drops when additional dimensions are included, which are learnt to model the PIE variations. In contrary, the identity features $\mathbf{y}^d$ can eliminate the dimensions that are not helpful for identity classification through the weight matrix $\mathbf{W}^d$, resulting in continuously improved performance w.r.t. n.

We further analyze how the energy vectors evolve over time during training. Specifically, at each epoch, we compute the energy vectors for each task. Then we compute the mean and standard deviation of each energy vector, as shown in Figure 3.6. Despite some local fluctuations, the overall trend is that the mean is decreasing and standard deviation is increasing as training goes on. This is because in the early stage of training, the energy vectors are more evenly distributed among all feature dimensions, which leads to the higher mean values and lower standard deviations. In the later stage of training, the energy vectors are shaped in a way to focus on some key dimensions for each task, which leads to the lower mean values and higher standard deviations.

The CNN learns to allocate a separate set of dimensions in the shared features to each task. The total number of dimensions assigned to each task depends on the loss weights. Recall that we obtain the overall loss weight for the side tasks as $\varphi_s = 0.1$ via brute-force search. Figure 3.7 shows the energy distributions with $\varphi_s = 0.2$ and $\varphi_s = 0.3$, which are compared to Figure 3.5 (a) where $\varphi_s = 0.1$. We have two observations. First, a larger loss weight for the side tasks leads to more dimensions being assigned to the side tasks. Second, the energies in $\mathbf{s}^d$ increase in order to compensate the fact that the dimensions assigned to the main task decrease. Therefore, we conclude that the loss weights control the energy distribution between different tasks.

Table 3.5: Performance comparison on LFW dataset.

Method	#Net	Training Set	Metric	Acc. $\pm$ Std. (%)
DeepID2 [125]	1	202,599 images of 10,177 subjects, private	Joint-Bayes	95.43
DeepFace [129]	1	4.4M images of 4,030 subjects, private	cosine	95.92 ± 0.29
CASIANet [150]	1	494,414 images of 10,575 subjects, public	cosine	96.13 ± 0.30
Wang et al. [137]	1	404,992 images of 10,553 subjects, public	Joint-Bayes	96.2 ± 0.9
Littwin and Wolf [86]	1	404,992 images of 10,553 subjects, public	Joint-Bayes	98.14 ± 0.19
MultiBatch [128]	1	2.6M images of 12K subjects, private	Euclidean	98.20
VGG-DeepFace [101]	1	2.6M images of 2,622 subjects, public	Euclidean	98.95
Wen et al. [141]	1	0.7M images of 17,189 subjects, public	cosine	99.28
FaceNet [115]	1	260M images of 8M subjects, private	L2	99.63 ± 0.09
s-CNN (ours)	1	494,414 images of 10,575 subjects, public	cosine	97.87 ± 0.70
m-CNN (ours)	1	494,414 images of 10,575 subjects, public	cosine	98.07 ± 0.57
p-CNN (ours)	1	494,414 images of 10,575 subjects, public	cosine	98.27 ± 0.64

3.3.3 Unconstrained Face Recognition

Experimental Settings We use CASIA-Webface [150] as our training set and evaluate on LFW, CFP, and IJB-A datasets. CASIA-Webface consists of 494,414 images of 10,575 subjects. LFW consists of 10 folders each with 300 same-person pairs and 300 different-person pairs. Given the saturated performance of LFW mainly due to its mostly frontal view faces, CFP and IJB-A are introduced for large-pose face recognition. CFP is composed of 500 subjects with 10 frontal and 4 profile images for each subject. Similar to LFW, CFP includes 10 folders, each with 350 same-person pairs and 350 different-person pairs, for both frontal-frontal (FF) and frontal-profile (FP) verification protocols. IJB-A dataset includes 5,396 images and 20,412 video frames of 500 subjects. It defines template-to-template matching for both face verification and identification.

In order to apply the proposed m-CNN and p-CNN, we need to have the labels for the side tasks. However, it is not easy to manually label our training set. Instead, we only consider pose estimation as the side task and use the estimated pose as the label for training. We use PIFA [69] to estimate 34 landmarks and the yaw angle, which defines three groups: right profile $[-90^\circ, -30^\circ)$, frontal $[-30^\circ, 30^\circ]$, and left profile $(30^\circ, 90^\circ]$. Figure 3.8 shows the distribution of the yaw angle

Table 3.6: Performance comparison on CFP dataset. Results reported are the average ± standard deviation over the 10 folds.

Method ↓	Frontal-Frontal			Frontal-Profile		
Metric (%) →	Accuracy	EER	AUC	Accuracy	EER	AUC
Sengupta et al. [116]	96.40 ± 0.69	3.48 ± 0.67	99.43 ± 0.31	84.91 ± 1.82	14.97 ± 1.98	93.00 ± 1.55
Sankarana. et al. [114]	96.93 ± 0.61	2.51 ± 0.81	99.68 ± 0.16	89.17 ± 2.35	8.85 ± 0.99	97.00 ± 0.53
Chen, et al. [22]	**98.67** ± 0.36	**1.40** ± 0.37	**99.90** ± 0.09	91.97 ± 1.70	8.00 ± 1.68	97.70 ± 0.82
DR-GAN [132]	97.84 ± 0.79	2.22 ± 0.09	99.72 ± 0.02	93.41 ± 1.17	6.45 ± 0.16	97.96 ± 0.06
Peng, et al. [102]	**98.67**	–	–	93.76	–	–
Human	96.24 ± 0.67	5.34 ± 1.79	98.19 ± 1.13	94.57 ± 1.10	5.02 ± 1.07	98.92 ± 0.46
s-CNN (ours)	97.34 ± 0.99	2.49 ± 0.09	99.69 ± 0.02	90.96 ± 1.31	8.79 ± 0.17	96.90 ± 0.08
m-CNN (ours)	97.77 ± 0.39	2.31 ± 0.06	99.69 ± 0.02	91.39 ± 1.28	8.80 ± 0.17	97.04 ± 0.08
p-CNN (ours)	97.79 ± 0.40	2.48 ± 0.07	99.71 ± 0.02	**94.39** ± 1.17	**5.94** ± 0.11	**98.36** ± 0.05

Table 3.7: Performance comparison on IJB-A.

Method ↓	Verification		Identification	
Metric (%) →	@FAR=0.01	@FAR=0.001	@Rank-1	@Rank-5
OpenBR [77]	23.6 ± 0.9	10.4 ± 1.4	24.6 ± 1.1	37.5 ± 0.8
GOTS [77]	40.6 ± 1.4	19.8 ± 0.8	44.3 ± 2.1	59.5 ± 2.0
Wang et al. [137]	72.9 ± 3.5	51.0 ± 6.1	82.2 ± 2.3	93.1 ± 1.4
PAM [95]	73.3 ± 1.8	**55.2** ± 3.2	77.1 ± 1.6	88.7 ± 0.9
DR-GAN [132]	77.4 ± 2.7	53.9 ± 4.3	85.5 ± 1.5	**94.7** ± 1.1
DCNN [21]	**78.7** ± 4.3	–	85.2 ± 1.8	93.7 ± 1.0
s-CNN (ours)	75.6 ± 3.5	52.0 ± 7.0	84.3 ± 1.3	93.0 ± 0.9
m-CNN (ours)	75.6 ± 2.8	51.6 ± 4.5	84.7 ± 1.0	93.4 ± 0.7
p-CNN (ours)	77.5 ± 2.5	53.9 ± 4.2	**85.8** ± 1.4	93.8 ± 0.9

estimation and the average image of each pose group. CASIA-Webface is biased towards frontal faces with 88% faces belonging to the frontal pose group based on our pose estimation.

The network structures are similar to those experiments on Multi-PIE. All models are trained from scratch for 15 epochs with a batch size of 8. The initial learning rate is set to 0.01 and reduced at the 10th and 14th epoch with a factor of 0.1. The other parameter settings and training process are the same as those on Multi-PIE. We use the same pre-processing as in [150] to align a face image. Each image is horizontally flipped for data augmentation in the training set. We also generate the mirror image of an input face in the testing stage. We use the average cosine distance

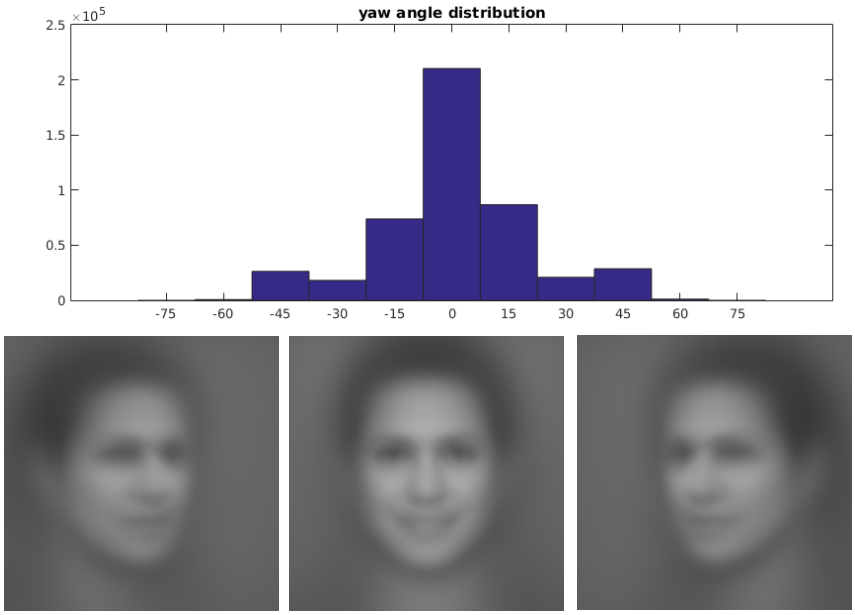


Figure 3.8: Yaw angle distribution on CASIA-Webface dataset.

of all four comparisons between the image pair and its mirror images for face recognition.

Performance on LFW Table 3.5 compares our face verification performance with state-of-the-art methods on LFW dataset. We follow the unrestricted with labeled outside data protocol. Although it is well-known that an ensemble of multiple networks can improve the performance [124, 126], we only compare CNN-based methods with one network for fair comparison. Our implementation of the CASIA-Net (s-CNN) with BN achieves much better results compared to the original performance [150]. Even with such a high baseline, m-CNN and p-CNN can still improve, achieving comparable results with state of the art, or better results if comparing to those methods trained with the same amount of data. Since LFW is biased towards frontal faces, we expect the improvement of our proposed m-CNN and p-CNN to the baseline s-CNN to be larger if they are tested on cross-pose face verification.

Performance on CFP Table 3.6 shows our face verification performance comparison with state-of-the-art methods on CFP dataset. For FF setting, m-CNN and p-CNN improve the verification

rate of s-CNN slightly. This is expected, as there is little pose variation. For FP setting, p-CNN substantially outperforms s-CNN and prior work, reaching close-to-human performance (94.57%). Note our accuracy of 94.39% is 9% relative error reduction of the previous state of the art [102] with 93.76%. Therefore, the proposed divide-and-conquer scheme is effective for in-the-wild face verification with large pose variation. And the proposed stochastic routing scheme improves the robustness of the algorithm. Even with the estimated pose serving as the ground truth pose label for MTL, the models can still disentangle the pose variation from the learnt identity features for pose-invariant face verification.

Performance on IJB-A We conduct close-set face identification and face verification on IJB-A dataset. First, we retrain our models after removing 26 overlapped subjects between CASIA-Webface and IJB-A. Second, we fine-tune the retrained models on the IJB-A training set of each fold for 50 epochs. Similar to [137], we separate all images into “well-aligned" and “poorly-aligned" faces based on the face alignment results and the provided annotations. In the testing stage, we only select images from the “well-aligned" faces for recognition. If all images in a template are “poorly-aligned" faces, we select the best aligned face among them. Table 3.7 shows the performance comparison on IJB-A. Similarly, we only compare to the methods with a single model. The proposed p-CNN achieves comparable performance in both face verification and identification.

3.4 Summary

This work explores multi-task learning for face recognition with PIE estimations as the side tasks. We propose a dynamic-weighting scheme to automatically assign the loss weights for each side task during training. MTL helps to learn more discriminative identity features by disentangling

the PIE variations. We further propose a pose-directed multi-task CNN with stochastic routing scheme to direct different paths for face images with different poses. We make the first effort to study face identification on the entire Multi-PIE dataset with full PIE variations. Extensive experiments on Multi-PIE show that our m-CNN and p-CNN can dramatically improve face recognition performance, especially on large poses. The proposed method is applicable to in-the-wild datasets with the estimated poses serving as the labels for training. We have achieved state-of-the-art performance on LFW, CFP, and IJB-A, showing the value of MTL for pose-invariant face recognition in the wild.

Chapter 4

Lage-Pose Face Frontalization

4.1 Introduction

This work presents Face Frontalization-Generative Adversarial Network (FF-GAN) to generate a frontal face from a face image with arbitrary pose while maintaining high quality and preserving identity. Face frontalization is the second category of methods for PIFR. It is motivated by the fact that comparing frontal faces is a much easier task than comparing faces under extreme profile views with self-occlusion. By filling the missing information, face frontalization has the potential to boost face recognition performance. Besides aiding recognition, frontalization of a face image is also a problem of independent interest, with potential applications such as face editing, accessorizing, and creation of models in virtual and augmented reality.

Synthesizing a frontal face from a single image with large pose variation is a challenging problem. A straight-forward method is to build a 3D model of the face and rotate the model to frontal view. Early work on face frontalization in computer vision rely on frameworks inspired by computer graphics. The well-known 3D Morphable Model (3DMM) [12] explicitly models facial shape and appearance to match an input image as close as possible. Subsequently, the recovered shape and appearance can be used to generate a face image under novel viewpoints. Many 3D face reconstruction methods [110, 166] build upon this direction by improving speed or accuracy. Deep learning has made inroads into data-driven estimation of 3DMM too [169, 72], circumventing

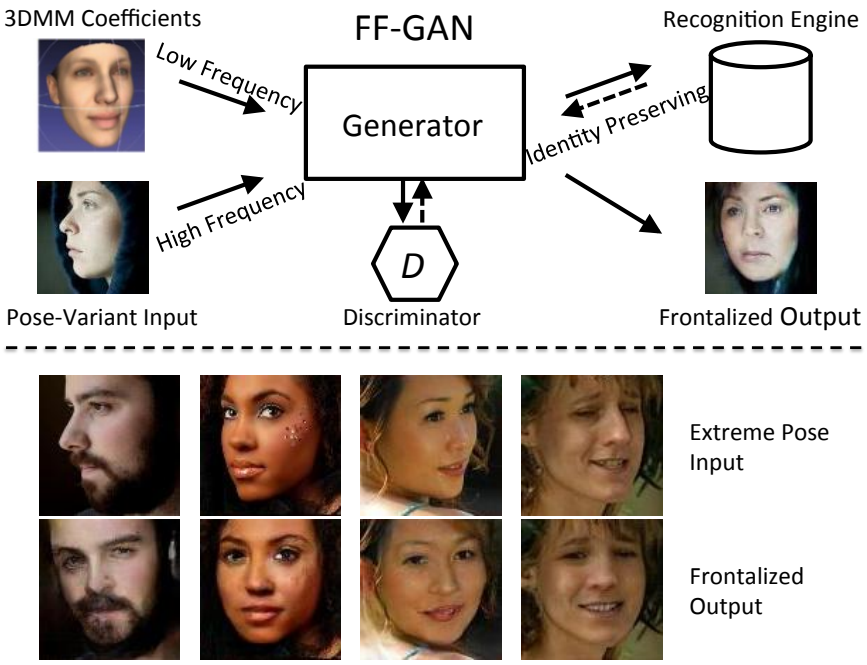


Figure 4.1: The proposed FF-GAN framework. Given a non-frontal face image as input, the generator produces a high-quality frontal face. Learned 3DMM coefficients provide global pose and low frequency information, while the input image injects high frequency local information. A discriminator distinguishes generated faces against real ones, where high-quality frontal faces are considered as real ones. A face recognition engine is used to preserve identity information. The output is a high quality frontal face that retains identity.

some drawbacks of early methods such as over-reliance on the accuracy of 2D landmark localization. Due to restricted Gaussian assumptions and nature of losses used, insufficient representation ability for facial appearance prevents such deep models from producing outputs of high quality. While inpainting methods such as [167] attempt to minimize the impact on quality due to self-occlusions, they still do not retain identity information.

In contrast, the proposed FF-GAN incorporates elements from both deep 3DMM and face recognition CNNs to achieve high-quality and identity-preserving frontalization, using a single input image that can be a profile view up to 90°. As shown in Figure 4.1, FF-GAN consists of four modules, the reconstructor, the generator, the discriminator, and the recognizer. The reconstruction module provides a useful prior to regularize the frontalization. However, it is well-known that deep

3DMM reconstruction is limited in the ability to retain high-frequency information. Therefore, the generation module combines both the 3DMM coefficients with the input image to generate a frontal face that maintains both global pose accuracy and retains local information present in the input image. In particular, the generator in FF-GAN produces a frontal image based on a reconstruction loss, a smoothness loss, and a novel symmetry-enforcing loss. The goal of the generator is to fool the discriminator into being unable to distinguish the generated frontal image from a real one. However, neither the 3DMM that loses high-frequency information, nor the GAN that only aligns domain-level distributions, suffice to preserve identity information in the generated image. To retain identity information, a recognition module is used to align the feature representation of the generated image with the input. A balanced training with all the above objectives results in high-quality frontalized faces that preserve identity.

To summarize, our key contributions are:

- A novel GAN-based end-to-end deep framework to achieve face frontalization even for extreme viewpoints.
- A deep 3DMM reconstruction module provides shape and appearance regularization beyond the training data.
- Effective symmetry-based loss and smoothness regularization that lead to the generation of high-quality images.
- Use of a deep face recognition CNN to enforce that the generated faces satisfy identity-preservation, besides realism and frontalization.
- Consistent improvements on several datasets across multiple tasks, such as face recognition, landmark localization, and 3D reconstruction.

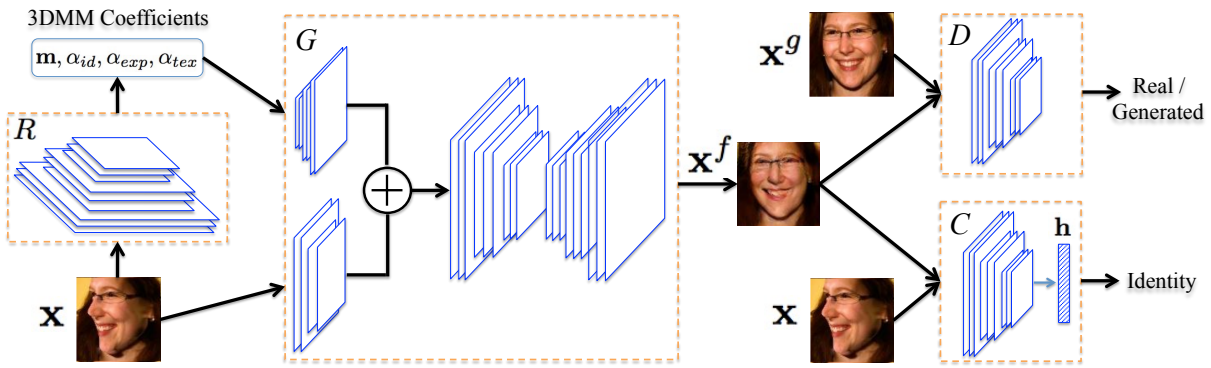


Figure 4.2: The proposed framework of FF-GAN. *R* is the reconstruction module for 3DMM coefficients estimation. *G* is the generation module to synthesize a frontal face. *D* is the discrimination module to make real or generated decision. *C* is the recognition module for identity classification.

4.2 Proposed Method

Figure 4.2 shows the framework of FF-GAN. The mainstay of FF-GAN is a generative adversarial network that consists of a generator *G* and a discriminator *D*. *G* takes a non-frontal face as input to generate a frontal output, while *D* attempts to classify it as a real frontal image or a generated one. Additionally, we include a face recognition engine *C* that regularizes the generator output to preserve identity features. A key component is a deep 3DMM module *R* that provides shape and appearance priors to the GAN. The reconstruction module *R* plays a crucial role in alleviating the difficulty of large pose face frontalization.

Let $\mathbb{D} = \{\mathbf{x}_i, \mathbf{x}_i^g, \mathbf{p}_i^g, y_i\}_{i=1}^N$ be the training set with N samples, with each sample consisting of an input image $\mathbf{x}_i$ with arbitrary pose, a corresponding ground truth frontal face $\mathbf{x}_i^g$, the ground truth 3DMM coefficients $\mathbf{p}_i^g$ and the identity label y_i. We henceforth omit the sample index i for clarity.

4.2.1 Reconstruction Module

Frontalization from extreme pose is a challenging problem. While a purely data-driven approach might be possible given sufficient data and an appropriate training regimen, however it is non-

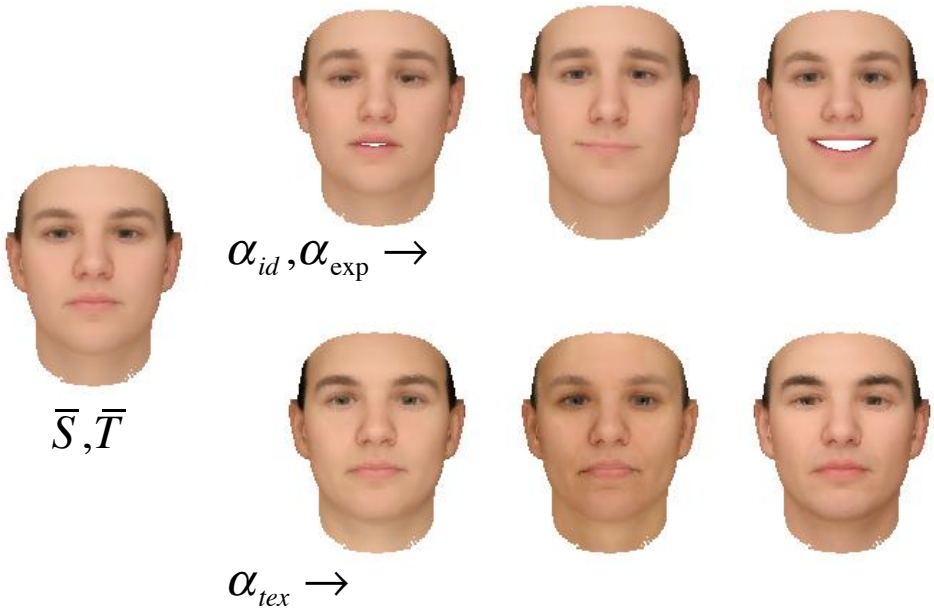


Figure 4.3: 3D faces generated with identity, expression, and texture variations.

trivial. Therefore, we propose to impose a prior on the generation process, in the form of a 3D Morphable Model (3DMM) [12]. This reduces the training complexity and leads to better empirical performance with limited data. Recall that 3DMM represents faces in the PCA space:

$$\begin{aligned} \mathbf{S} &= \bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}, \\ \mathbf{T} &= \bar{\mathbf{T}} + \mathbf{A}_{tex}\alpha_{tex}, \end{aligned} \tag{4.1}$$

where $\mathbf{S}$ represent the 3D shape coordinates computed as the linear combination of the mean shape $\bar{\mathbf{S}}$, the shape basis $\mathbf{A}_{id}$, and the expression basis $\mathbf{A}_{exp}$, while $\mathbf{T}$ is the texture (RGB color values) that is the linear combination of the mean texture $\bar{\mathbf{T}}$ and the texture basis $\mathbf{A}_{tex}$.

The coefficients $\{\alpha_{id}, \alpha_{exp}, \alpha_{tex}\}$ defines a unique 3D face. Figure 4.3 shows some examples generated by varying the shape coefficients (α_{id} and α_{exp}) or the texture coefficients α_{tex}. The identity coefficients change the structure of the face via the expression coefficients mainly change the mouth region of the face. The texture coefficients change the appearance of the face.

The 3D shape $S \in \mathbb{R}^{3\times Q}$ records the x, y, z coordinates of Q vertexs on the 3D face model,

$$\mathbf{S} = \begin{bmatrix} x_1 & x_2 & \dots & x_Q \\ y_1 & y_2 & \dots & y_Q \\ z_1 & z_2 & \dots & z_Q \end{bmatrix}. \tag{4.2}$$

Let $U \in \mathbb{R}^{2\times Q}$ denote the corresponding x, y coordinates on the 2D face image,

$$\mathbf{U} = \begin{bmatrix} u_1 & u_2 & \dots & u_Q \\ v_1 & v_2 & \dots & v_Q \end{bmatrix}. \tag{4.3}$$

In order to build the correspondence between the 2D shape $\mathbf{U}$ with the 3D shape $\mathbf{S}$, the camera projection parameters are needed. Previous work [68, 166] have applied 3DMM for face alignment where a weak perspective projection model is used to project the 3D shape into the 2D space. Similar to [68], we calculate a projection matrix $\mathbf{m} \in \mathbb{R}^{2\times 4}$ based on pitch, yaw, roll, scale and 2D translations so that $\mathbf{U} = \mathbf{m}[\mathbf{S}; \mathbf{1}]$ ($\mathbf{1}$ represents a vector of 1s being concatenated to $\mathbf{S}$ for matrix multiplication).

Let $\mathbf{p} = \{\mathbf{m}, \alpha_{id}, \alpha_{exp}, \alpha_{tex}\}$ denotes the 3DMM coefficients. The target of our reconstruction module R is to estimate $\mathbf{p} = R(\mathbf{x})$ given an input image $\mathbf{x}$. Since the intent is for R to also be trainable with the rest of the framework, we use a CNN model based on CASIA-Net [150] for this regression task. We apply z-score normalization to each dimension of the parameters before training. A weighted parameter distance cost similar to [166] is used:

$$\min_{\mathbf{p}} L_R = (\mathbf{p} - \mathbf{p}^g)^\top \mathbf{W} (\mathbf{p} - \mathbf{p}^g), \tag{4.4}$$

where $\mathbf{W}$ is the importance matrix whose diagonal is the weight of each parameter. The weight is

calculated based on the 2D landmark errors caused by the error in the estimation of each parameter. **W** is calculated once and kept the same during training.

4.2.2 Generation Module

The pose estimation obtained from module R is quite accurate. However, the shape and texture coefficients estimations lead to the loss of high frequency details presented in the original image. This is understandable since a low-dimensional PCA representation can preserve most of the energy with lower frequency components. Thus, we use a generative module that relies on the 3DMM coefficients $\mathbf{p}$ and the input image $\mathbf{x}$ to recover a frontal face that preserves both the low and high frequency components. Our generator relies on multiple objectives for the frontalization task as described below respectively.

In Figure 4.2, features from the two inputs to the generator G are fused through an encoder-decoder network to synthesize a frontal face $\mathbf{x}^f = G(\mathbf{x}, \mathbf{p})$. To penalize the generated output from the ground truth frontal face $\mathbf{x}^g$, one straight-forward objective is the reconstruction loss that aims at reconstructing the ground truth with minimal error:

$$L_{G_{rec}} = \|G(\mathbf{x}, \mathbf{p}) - \mathbf{x}^g\|_1. \tag{4.5}$$

Since an L_2 loss empirically leads to blurry output, we use an L_1 loss instead to better preserve high frequency component. At the beginning of training, the reconstruction loss harms the overall process since the generation is far from frontalized, so the reconstruction loss operates on a poor set of correspondences. Thus, the weight for the reconstruction loss should be set in accordance to the training stage. The details of tuning the weight are discussed in Section 4.3.2.

To reduce block artifacts, we use a spatial total variation loss to encourage smoothness in the

generated output:

$$L_{G_{tv}} = \frac{1}{|\Omega|} \int_{\Omega} |\nabla G(\mathbf{x}, \mathbf{p})| du, \tag{4.6}$$

where $|\nabla G|$ is the image gradient, $u \in \mathbb{R}^2$ is the two dimensional coordinate increment, Ω is the image region, and $|\Omega|$ is the area normalization factor.

Based on the observation that human faces share self-similarity across left and right halves, we explicitly impose a symmetry loss. As shown in Figure 4.4, we recover a frontalized 2D projected mask $\mathscr{M}$ from the frontalized 3DMM coefficients indicating the visible parts of the face. The mask $\mathscr{M}$ is binary, with nonzero values indicating the visible regions and zero otherwise. By horizontally flipping the face, we can generate another mask $\mathscr{M}_{flip}$ indicating the visible region of the flipped input image. We demand that the generated frontal face for the original input image and its flipped version should be similar within their respective masks:

$$\begin{aligned} L_{G_{sym}} &= \|\mathscr{M} \odot G(\mathbf{x}, \mathbf{p}) - \mathscr{M} \odot G(\mathbf{x}_{flip}, \mathbf{p}_{flip})\|_2 \\ &+ \|\mathscr{M}_{flip} \odot G(\mathbf{x}, \mathbf{p}) - \mathscr{M}_{flip} \odot G(\mathbf{x}_{flip}, \mathbf{p}_{flip})\|_2. \end{aligned} \tag{4.7}$$

Here, $\mathbf{x}_{flip}$ is the horizontally flipped image for the input image $\mathbf{x}$, $\mathbf{p}_{flip}$ (only the pose parameters $\mathbf{m}$ is changed the remaining is the same) are the 3DMM coefficients for $\mathbf{x}_{flip}$ and $\odot$ denotes the element-wise multiplication. We emphasize on the mask because those invisible parts during rotation may not be confident to contribute to the penalty, whereas the role of the mask is to focus on the visible parts for both the original image and the flipped image, rather than the background.

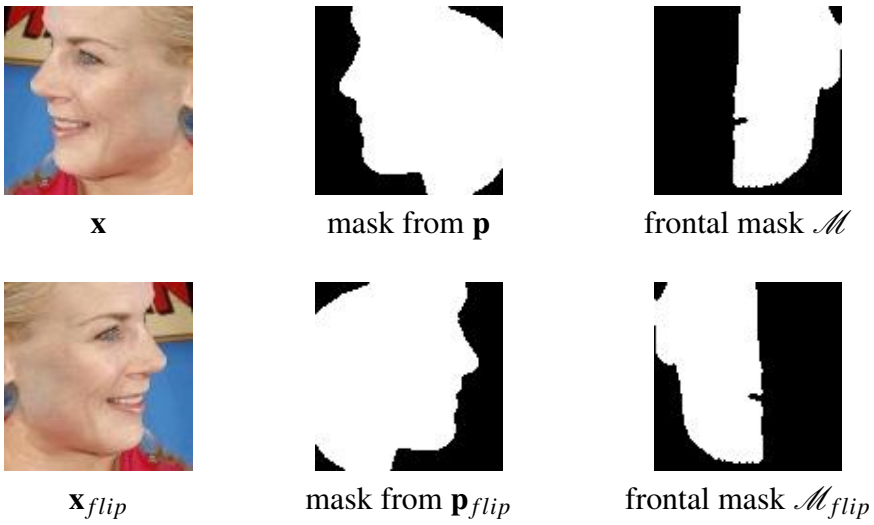


Figure 4.4: Image flip and mask generation process for the symmetry loss.

4.2.3 Discrimination Module

Generative Adversarial Network (GAN) [44], formulated as a two-player minimax game between a generator G and a discriminator D, has been widely used for image generation [35]. In this work, G synthesizes a frontal face image $\mathbf{x}^f$ and D distinguishes between the generated face from the real frontal face $\mathbf{x}^g$. Note that in a conventional GAN, all images used for training are considered as real samples. However, we limit the definition of "real" sample to be the face images with frontal view only. Therefore, G is trained to not only generate realistic but also frontal face images.

The discriminator D consists of five convolution layers and one linear layer that generates a 2D vector with each dimension representing the probability of the input to be real or generated. During training, D is updated with two batches of samples in each iteration. The following objective is maximized:

$$\min_D L_D = \mathbb{E}_{\mathbf{x}^g \in \mathscr{R}} \log(D(\mathbf{x}^g)) + \mathbb{E}_{\mathbf{x} \in \mathscr{K}} \log(1 - D(G(\mathbf{x}, \mathbf{p}))) \tag{4.8}$$

where $\mathscr{R}$ and $\mathscr{K}$ are the real and generated image sets respectively.

On the other hand, G aims to fool D to classify the generated image $G(\mathbf{x}, \mathbf{p})$ to be real with the

following loss:

$$L_{G_{gan}} = \mathbb{E}_{\mathbf{x}\in\mathscr{K}} \log(D(G(\mathbf{x},\mathbf{p}))) \tag{4.9}$$

The competition between G and D improves both modules. In the early stages when face images are not fully frontalized, D focuses on the pose of the face to make the real or generated decision, which in turn helps G to generate a frontal face. In the later stages when face images are frontalized, D focuses on subtle details of frontal faces, which guides G to generate a realistic frontal face that is difficult to achieve with the supervisions of (4.5), (4.6) and (4.7).

4.2.4 Recognition Module

A key challenge in large-pose face frontalization is to preserve the original identity in the generated frontal face. This is a difficult task due to self-occlusion in profile faces. The above discriminator can only determine whether the generated image is realistic and in frontal view but cannot tell whether the identity of the input image is retained. Although we have L1, total variation, and masked symmetry losses for face generation, they treat each pixel equally that result in the loss of discriminative power for the identity features. Therefore, we use a recognition module C to impart correct identity to the generated images.

C is a general face recognition engine that any state-of-the-art framework can be easily plugged in. We use a CASIA-Net structure, which has proved to work well for face recognition. A cross-entropy loss is used for training C to classify image $\mathbf{x}$ with the ground truth identity $\mathbf{y}$. Here $\mathbf{y}$ is a one-hot vector with the element of the correct identity to be 1.

$$\min_C L_C = \sum_j \left[-y_j \log(C_j(\mathbf{x})) - (1-y_j)\log(1-C_j(\mathbf{x}))\right], \tag{4.10}$$

where j is the index of the identity classes. $C_j(\mathbf{x})$ is the probability of the input $\mathbf{x}$ belonging to the jth identity.

Similar to the competition between G and D. Now, our generator G must also fool C to classify the generated image to have the same identity as the input image. If the identity label of the input image is not available, we regularize the extracted identity features $\mathbf{h}^f$ of the generated image to be similar to those of the input image, denoted as $\mathbf{h}$. During training, C is updated with real input images to retain discriminative power. The loss from the generated images is back-propagated to update the generator G:

$$L_{G_{id}} = \begin{cases} -\log(C(G(\mathbf{x},\mathbf{p}))), & \exists y \\ \|\mathbf{h}^f - \mathbf{h}\|_2^2, & \nexists y. \end{cases} \tag{4.11}$$

To summarize the framework, the reconstruction module R provides 3DMM prior knowledge to the frontalization process through (4.4), the discriminator D does so through (4.8) and the recognition engine C through (4.10). The generator G combines all these sources of information to optimize an overall objective function:

$$\min_G L_G = \lambda_{rec} L_{G_{rec}} + \lambda_{tv} L_{G_{tv}} + \lambda_{sym} L_{G_{sym}} + \lambda_{gan} L_{G_{gan}} + \lambda_{id} L_{G_{id}}. \tag{4.12}$$

It is important to balance the weights between each loss, which are discussed in Section 4.3.2 to illustrate how each component contributes to the joint optimization of G.

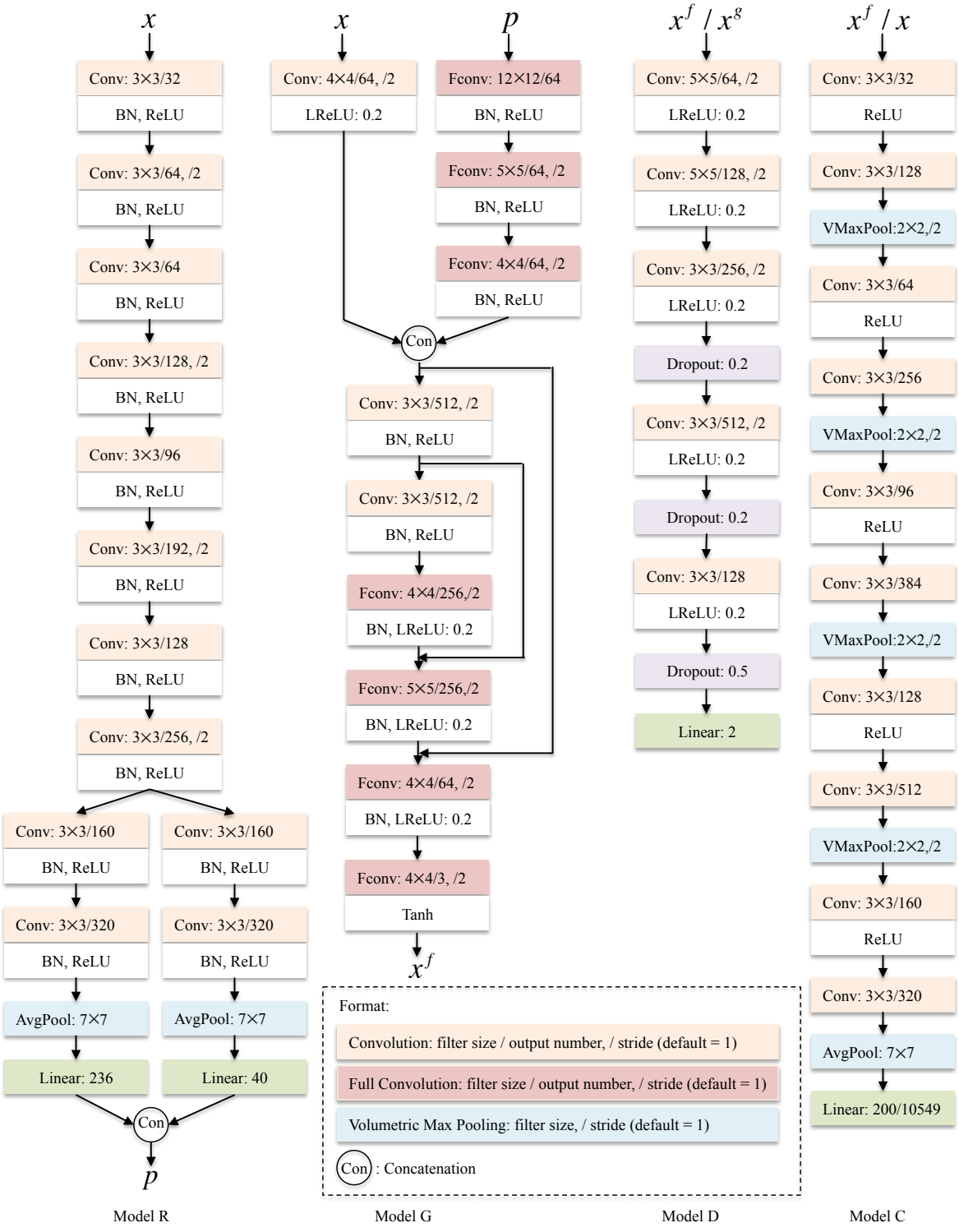


Figure 4.5: Detailed network structure of FF-GAN.

4.3 Implementation Details

4.3.1 Network Structures

Figure 4.5 shows the detailed network structure of FF-GAN, composed of the 3DMM reconstruction module R, the generator G, the discriminator D, and the recognition engine C.

The 3DMM module R takes the input image $\mathbf{x}$ and generates the 3DMM coefficients $\mathbf{p}$ including the weak perspective matrix $\mathbf{m} \in \mathbb{R}^{8\times1}$, the shape coefficients $\alpha_{id} \in \mathbb{R}^{199\times1}$, the expression coefficients $\alpha_{exp} \in \mathbb{R}^{29\times1}$, and the texture coefficients $\alpha_{tex} \in \mathbb{R}^{40\times1}$. We use the provided coefficients in [166] as our ground truth for training. Originally, 3DMM consists of 199 bases for texture model. Only the first 40 bases are used in [166]. We use the CASIA-Net structure, where the texture coefficients are separated from the shape-related coefficients in the later layers, which empirically demonstrates better performance in our experiments.

The generator G takes the image $\mathbf{x}$ and the estimated 3DMM coefficients $\mathbf{p}$ as the inputs to generate a frontal-view face $\mathbf{x}^f$. The 3DMM coefficients provide a frontal low frequency basis and the detailed appearance is expected to be recovered from the raw pose-variant input image. Clearly, these two inputs are not in the same domain. We apply three fully convolution layers to up-sample $\mathbf{p}$ and one convolution layer to down-sample $\mathbf{x}$ to the same size of $50 \times 50 \times 64$. The outputs are concatenated to an encoder-decoder structured network which includes two skip connections that are used to provide high frequency information to the decoding process. The feature after encoding is of dimension $512 \times 12 \times 12$ which maintains the spatial information to recover the input image when necessary, i.e., if the input image $\mathbf{x}$ is already of frontal view, our network should produce an identity mapping.

The discriminator D aims to distinguish between the generated face $\mathbf{x}^f$ and the real frontal face $\mathbf{x}^g$. This is a relatively easy task, so we use a shallow network with five convolution layers and one

linear layer, which outputs a 2D vector with each dimension indicating the probability of the input belonging to the generated image or the real image. In each iteration during training, D is updated with two batches of samples from $\mathbf{x}^f$ and $\mathbf{x}^g$, respectively.

The recognition engine C also adopts a CASIA-Net structure. Instead of using the max pooling layer as CASIA-Net, we choose volumetric max pooling, which applies pooling not only in the spatial dimensions but also across the feature channels. We find this to be helpful for face recognition. C is pre-trained with CASIA-Webface dataset [150] and fixed in the first two stages of the training process. Later, we update C using the original input image $\mathbf{x}$. Note that $\mathbf{x}^f$ are the input to fool C during the training of G and gradients flow through C to update the generator G.

4.3.2 Training Strategies

Our framework consists of mainly four parts as shown in Figure 4.2, the deep 3DMM reconstructor R, a two-way fused encoder-decoder generator G, the real/generated discriminator D and a face recognition engine C jointly trained for the identity regularization. The training of the overall network can be hardly initialized from scratch. The generator G expects to receive the correct 3DMM coefficients, whereas the reconstructor R needs to be pre-trained. Our identity regularization also requires correct identity information from the recognizer. Thus, the reconstructor R is pre-trained until we achieve comparable performance for face alignment compared to previous work [166] using 300W-LP [166]. The recognizer is pre-trained using CASIA-Webface and verified with promising verification accuracy on LFW.

The end-to-end joint training is conducted after R and C are well pre-trained. Notice that we leave the generator G and the discriminator D training from scratch simultaneously because we believe pre-trained G and D do not contribute much to the adversarial training process. Good G with poor D will quickly pull G to be poor again and vice versa. Further these two components

should also match with each other. Good G may be evaluated poor by a good D as the discriminator may be trained from other sources.

4.4 Experimental Results

4.4.1 Settings and Datasets

We evaluate our proposed FF-GAN on a variety of tasks including face frontalization, landmark localization, 3D face reconstruction, and face recognition. Frontalization and 3D reconstruction are evaluated qualitatively by comparing the visual quality of the generated images to the ground truth. We also report some quantitative results on sparse 2D landmark localization accuracy, which indicates our method does fairly well on pose estimation, even though we do not train for this specific task. Face recognition is evaluated quantitatively over several challenging face verification and identification datasets. We pre-process the images by applying state-of-the-art face detection and face alignment algorithms and crop to 100×100 size across all the datasets. The face datasets used in this work are introduced below.

300W-LP consists of $122,450$ images that are augmented from 300W [113] by the face profiling approach of Zhu et al. [166], which is designed to generate images with yaw angles ranging from -90° to 90°. We use 300W-LP as our training set by forming image pairs of pose-variant and frontal-view images with the same identity. The estimated 3DMM coefficients provided with the images are treated as the ground truth to train module R.

AFLW2000 is constructed for 3D face alignment evaluation by the same face profiling method applied in 300W-LP. The dataset includes the estimated 3DMM coefficients and augmented 68 landmarks for the first $2,000$ images in AFLW. We use this dataset to evaluate module R for reconstruction.

Multi-PIE consists of 754,200 images from 337 subjects with large variations in PIE. We select a subset of 301,600 images with 13 poses, 20 illuminations, neutral expression from all four sessions. The first 200 subjects are used for training and the remaining 137 subjects for testing, similar to the setting of [132]. We randomly choose one image for each subject with frontal pose and neutral illumination as gallery and all the rest as probe images.

CASIA-Webface consists of 494,414 images of 10,575 subjects where the images of 26 overlapping subjects with IJB-A are removed. It is a widely applied large-scale dataset for face recognition. We apply it to pre-train and finetune module C.

LFW contains 13,233 images collected from the Internet. The verification set consists of 10 folders, each with 300 same-person pairs and 300 different-person pairs. We evaluate face verification performance on frontalized images and compare with previous frontalization algorithms on LFW.

IJB-A includes 5,396 images and 20,412 video frames for 500 subjects, which is a challenging dataset with large pose variation. Different from previous datasets, IJB-A defines face template matching where each template contains a variant number of images. It consists of 10 folders, each of which being a different partition of the full set. We finetune model C on the training set of each folder and evaluate on the testing set for face verification and identification.

CFP is composed of 500 subjects with 10 frontal and 4 profile faces for each subject. We use this dataset to explore the frontalization quality of face images with extreme profile pose (90°).

For in-the-wild experiments, we train our model using 300W-LP. We prepare the training image pairs by setting one pose-variant face image (15°-90°) as the input and the frontal-view face image of the same subject (0°-15°) as the target. We use Adam solver for optimization with a batch size of 128. The weight decay is set to $2e{-}4$ and momentum is set to 0.9. The initial learning rate is set to $2e{-}4$. We reduce the learning rate by a factor of 10 for every 20 epochs.

As shown in (4.12), we set up five balance factors to control the contribution of each objective

to the overall loss. The end-to-end training can be divided into three stages. For the first stage, λ_{rec} is set to 0 and λ_{id} is set to 0.01, since these two parts are highly related with the mapping from the generated output to the reference input. Typical values for λ_{tv}, λ_{sym}, and λ_{gan} are all 1.0s. Once the training error of G and D strikes a balance within usually 20 epochs, we change λ_{rec} and λ_{id} to be 1.0s while tuning down λ_{tv} to be 0.5, λ_{sym} to be 0.8, respectively for the second stage. It takes another 20 epochs to strike a new balance. Notice that for these two stages' training, we fix model C. After that, we relax model C and further fine-tune all the modules jointly with a learning rate of $1e-6$ for the third stage.

For controlled experiments on Multi-PIE, we finetune the network from the models trained on 300W-LP. We mix the dataset of 300W-LP with Multi-PIE where 300W-LP is only used to update module R. The weights for each loss are set to 1. Since we already have a good starting point, we do not need to adjust the weights dynamically on Multi-PIE. The initial learning rate is set to $1e-4$ for the first two stages when model C is fixed and reduced to $5e-5$ when model C is relaxed. The first two stages need approximately 10 epochs for finetuning. The other hyper-parameters are the same as the experiments on 300W-LP.

During the testing stage, module R is used to estimate the 3DMM coefficients. Module G is used for face frontalization. Module C is used for feature extraction. Module D is used to predict the confidence score of the generated image. These outputs are all used in our experiments.

4.4.2 3D Reconstruction

FF-GAN borrows prior shape and appearance information from 3DMM to serve as the reference for frontalization. Though we do not specifically optimize for the reconstruction task, it is interesting to see whether our reconstructor is doing a fair job in the 3D reconstruction task.

Figure 4.6 (a) shows five examples on AFLW2000 for landmark localization and frontaliza-

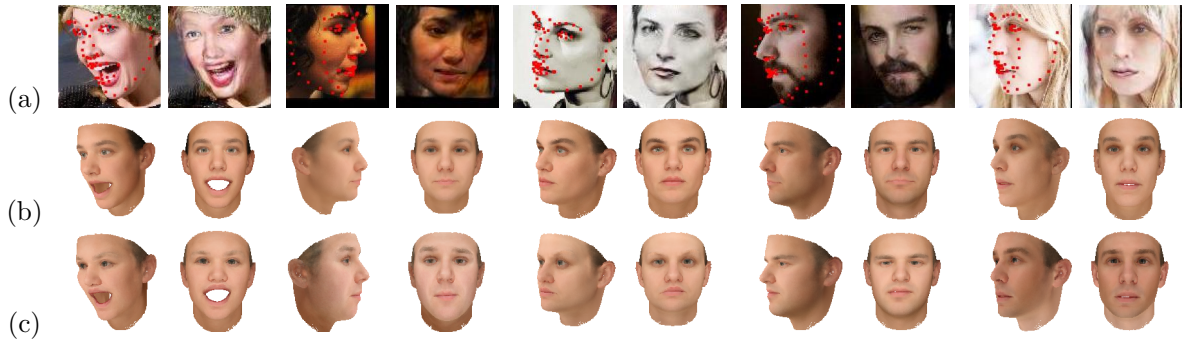


Figure 4.6: (a) Our landmark localization and face frontalization results; (b) Our 3DMM estimation; (c) Ground truth from [166].

tion. Our method localizes the key points correctly and generates realistic frontal faces even for extreme profile inputs. We also quantitatively evaluated the landmark localization performance using the normalized mean square error. Our model R achieves 6.01 normalized mean square error, compared to 5.42 for 3DDFA [166] and 6.12 for SDM [146]. Note that our method achieves competitive performance compared to 3DDFA and SDM, even though those methods are tuned specifically for the localization task. This indicates that our reconstruction module performs well in providing correct geometric information.

Given the input images in (a), we compute the 3DMM coefficients with our model R and generate the 3D geometry and texture using (4.1), as shown in Figure 4.6 (b). We observe that our method effectively preserves shape and identity information in the estimated 3D models, which can even outperform the ground truth provided by 3DDFA. For example, the shape and texture estimations in the last example is more similar to the input while the ground truth clearly shows a male subject rather than a female. Given that the 3DMM coefficients cannot preserve local appearance, we obtain such high frequency information from the input image. Thus, the choice of fusing 3DMM coefficients with the original input is shown to be a reasonable one empirically.

4.4.3 Face Recognition

One of our motivation for face frontalization is to see, whether the frontalized images bring in the correct identity information for the self-occlusion missing part, and thus boost the performance in face recognition. To verify this, we evaluate our framework on LFW [62], MultiPIE [45], and IJB-A [77] for verification and identification tasks. Features are extracted from module *C* across all the experiments. Euclidean distance is used as the metric for face matching.

Evaluation on LFW We evaluate the face verification performance on our frontalized images of LFW, compared to previous face frontalization methods. LFW-FF-GAN represents our method to generate frontalized images, LFW-3D is from [55] and LFW-HPEN from [167]. Those collected datasets are pre-processed in *the same way* as ours. Table 4.1 shows the face verification performance where the average and standard deviation over 10 folders are reported. Our method achieves strong results compared to the state-of-the-art methods, which verifies that our frontalization technique preserves the identity information.

Figure 4.7 shows some visual examples. Compared to the state-of-the-art face frontalization algorithms, the proposed FF-GAN can generate realistic and identity-preserved faces especially for large poses. The facial detail filling technique proposed in [167] relies on a symmetry assumption and may lead to inferior results (3rd row, 2nd and 7th column). In contrast, we introduce a symmetry loss in the training process that can generalize well to the test images without the need for post-processing to impose symmetry as a hard constraint.

Evaluation on IJB-A We further evaluate our algorithm on IJB-A dataset. Following prior work [132], we select a subset of well aligned images in each template for face matching.

We define our distance metric as the original image pair distance plus the weighted generated image pair distance. The weights are the confidence score provided by our module *D*, i.e.

Table 4.1: Performance comparison on LFW dataset with accuracy (ACC) and area-under-curve (AUC).

Dataset	ACC(%)	AUC(%)
Ferrari et al. [40]	-	94.29
LFW-3D [55]	93.62 ± 1.17	98.36 ± 0.06
LFW-HPEN [167]	96.25 ± 0.76	99.39 ± 0.02
LFW-FF-GAN	$\mathbf{96.42} \pm 0.89$	$\mathbf{99.45} \pm 0.03$

Figure 4.7: Face frontalization results comparison on LFW. (a) Input; (b) LFW-3D [55]; (c) HPEN [167]; (d) FF-GAN.

$D(G(\mathbf{x}, \mathbf{p}))$. Recall that module D is trained for the real or generated classification task, which reflects the quality of the generated images. Obviously, the poorer quality of the generated images, the less likely we take the generated image pair for the distance metric fusion. With the fused metric distance, we expect the generated images to provide complimentary information to boost the recognition performance.

Table 4.2 shows the verification and identification performance. On verification, our method achieves consistently better accuracy compared to the baseline methods. The gap is 6.46% at FAR 0.01 and 11.13% at FAR 0.001, which is a significant improvement. On identification, our fused metric also achieves consistently better result, 4.95% improvement at Rank-1 and 1.66% at Rank-

5. As a challenging face dataset in the wild, large pose variation, complex background, and the uncontrolled illumination prevents the compared methods to perform well. Closing one of those variation gaps would lead to large improvement, as evidenced by our face frontalization method in rectifying the pose variation.

Table 4.2: Performance comparison on IJB-A dataset.

Method ↓	Verification		Identification	
Metric (%) →	@FAR=0.01	@FAR=0.001	@Rank-1	@Rank-5
OpenBR [77]	23.6 ± 0.9	10.4 ± 1.4	24.6 ± 1.1	37.5 ± 0.8
GOTS [77]	40.6 ± 1.4	19.8 ± 0.8	44.3 ± 2.1	59.5 ± 2.0
Wang et al. [137]	72.9 ± 3.5	51.0 ± 6.1	82.2 ± 2.3	93.1 ± 1.4
PAM [95]	73.3 ± 1.8	55.2 ± 3.2	77.1 ± 1.6	88.7 ± 0.9
DCNN [21]	78.7 ± 4.3	–	85.2 ± 1.8	93.7 ± 1.0
DR-GAN [132]	77.4 ± 2.7	53.9 ± 4.3	85.5 ± 1.5	94.7 ± 1.1
FF-GAN	$\mathbf{85.2} \pm 1.0$	$\mathbf{66.3} \pm 3.3$	$\mathbf{90.2} \pm 0.6$	$\mathbf{95.4} \pm 0.5$

Evaluation on Multi-PIE Multi-PIE allows for a graded evaluation with respect to PIE variations. Thus, it is an important dataset to validate the performance of our methods with respect to prior work. The rank-1 identification rate is reported in Table 4.3. Note that previous works only consider poses within 60°, while our method can handle all pose variations including profile views at 90°. The results suggest that when pose variation is within 15°, which is near frontal, our method is competitive to state-of-the-art methods. But when the pose is 30° or larger, our method demonstrates significant advantages over all the other methods. We achieve 3.8% improvement at 60° and 2.8% better on the average accuracy from 0° to 60° compared to the previous best result. The average accuracy on 0° to 90° only drops 4.5% from our method's average on 0° to 60°. Further visual results in Figure 4.8 and 4.12, second row, also support that our method is almost invariant to pose.

Table 4.3: Performance comparison on Multi-PIE dataset.

	0^o	15^o	30^o	45^o	60^o	75^o	90^o	Avg(0^o-60^o)	Avg(0^o-90^o)
Zhu et al. [168]	94.3	90.7	80.7	64.1	45.9	–	–	72.9	–
Zhu et al. [169]	95.7	92.8	83.7	72.9	60.1	–	–	79.3	–
Yim et al. [151]	**99.5**	**95.0**	88.5	79.9	61.9	–	–	83.3	–
DR-GAN [132]	97.0	94.0	90.1	86.2	83.2	–	–	89.2	–
FF-GAN	95.5	94.8	**93.4**	**91.0**	**87.0**	**82.7**	**71.7**	**92.0**	87.5

4.4.4 Face Frontalization

In this section, we will illustrate further face frontalization results on Multi-PIE, AFLW, IJB-A, and CFP datasets.

Visualization on Multi-PIE Figure 4.8 shows the face frontalization results of eight subjects in the test set of Multi-PIE. The proposed FF-GAN generates realistic frontal faces that are similar to the ground truth (top rows are the input, where the frontal ground truth is the image in the middle column) across all different poses. Furthermore, the gender, race, and attributes like eyeglasses are well-preserved. It is clear that the larger the pose angle is, the more difficult it is for the generated output to preserve identity. Surprisingly, for large poses (up to 90°), FF-GAN can still preserve the identity to a large extent. To the best of our knowledge, this is the first work to show face frontalization results for faces beyond 60°.

Visualization on AFLW Figure 4.9 shows the face frontalization results on AFLW, which encompasses more pose variation than LFW. For better visualization, we separate the faces into three different groups with small, medium, and large pose variation, which are defined based on the visibility of the two eyes (both visible for small pose, one eye half-occluded for medium pose and one eye fully-occluded for large pose). FF-GAN works extremely well for the face images with small pose, in rows (a) and (b). For face images with medium or large poses in rows (c) and (d), respectively, FF-GAN still generates plausible results without many artifacts. We note that even

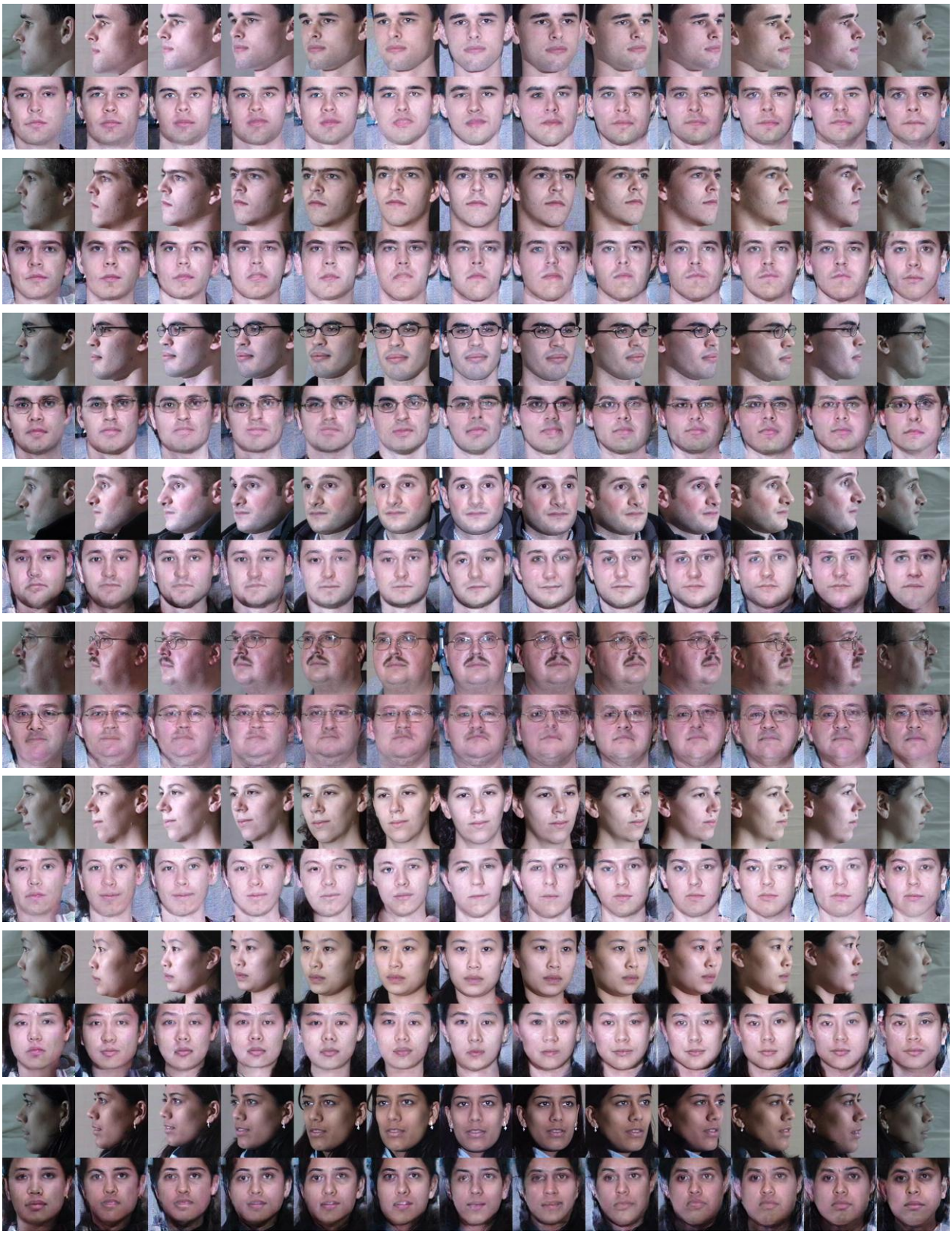


Figure 4.8: Visual results on Multi-PIE. Each example shows 13 pose-variant inputs (top) and the generated frontal outputs (bottom). We clearly observe that the outputs consistently recover similar frontal faces across all the pose intervals.

for nearly profile views in row (d), high-frequency details of facial features are recovered well, the frontalized face is symmetric and identity is preserved quite well. Row (e) shows results for input images under various lighting or expressions. Again FF-GAN works well under these variations.

Visualization on IJB-A Figure 4.10 shows the face frontalization results on IJB-A, which consists of large-pose and low-quality face images. The input images are of medium to large pose and under a large variation of race, age, expression, and lighting conditions. However, FF-GAN can still generate realistic and identity-preserved frontal faces.

Visualization on CFP We further explore face frontalization on CFP, which is a challenging dataset with extreme profile faces (90°). Note that our training set, 300W-LP, has a large systematic gap from CFP. Therefore, we finetune our models with a limited subset ($1,600$ profile and $4,000$ frontal images of 400 subjects) from CFP. Figure 4.11 shows the face frontalization results on some of the remaining unseen profile faces. Despite some artifacts and blurring effects in the occluded side of the face, FF-GAN manages to generate a consistent frontal face. We observe that identity is preserved to some degree and facial features are reconstructed to a reasonable extent. However, we observe that there is some blurriness in the frontalized output. This is attributed to the fact that the face images and crops are different from other datasets. For instance, the ear and neck regions are prominently visible in CFP but not in other datasets. Thus, they are not entirely eliminated in the frontalized output and cause ghosting artifacts. A larger training dataset that includes profile faces similar to those in CFP will likely alleviate this issue.

In summary, our frontalization results are of very high quality in Multi-PIE, LFW, AFLW, IJB-A datasets, with some room for improvement in the CFP dataset.

Figure 4.9: Face frontalization results on AFLW. FF-GAN achieves very promising visual effects for faces with small (row (a) and (b)), medium (row (c)), large (row (d)) poses and under various lighting conditions and expressions (row (e)). We observe that the proposed FF-GAN achieves accurate frontalization, while recovering high frequency facial details as well as identity, even for face images observed under extreme variations in pose, expression or illumination.

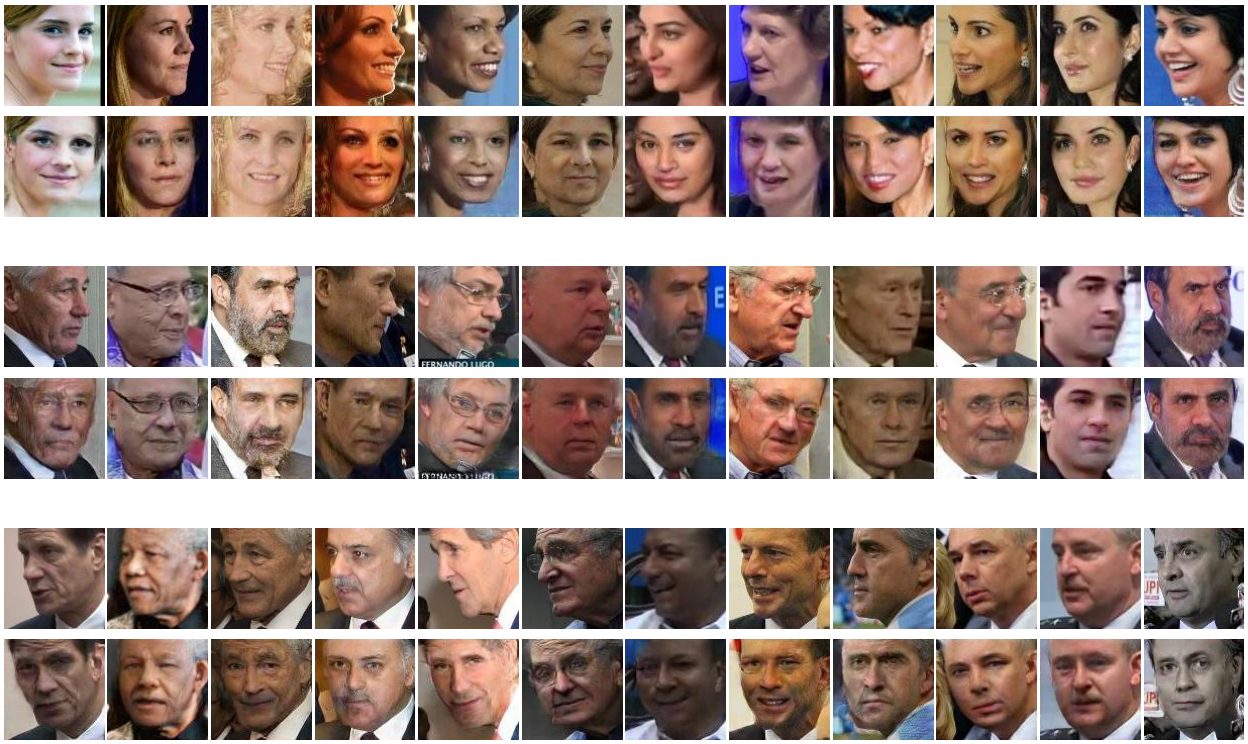


Figure 4.10: Face frontalization results on IJB-A. Odd rows are all profile-view inputs and even rows are the frontalized results.

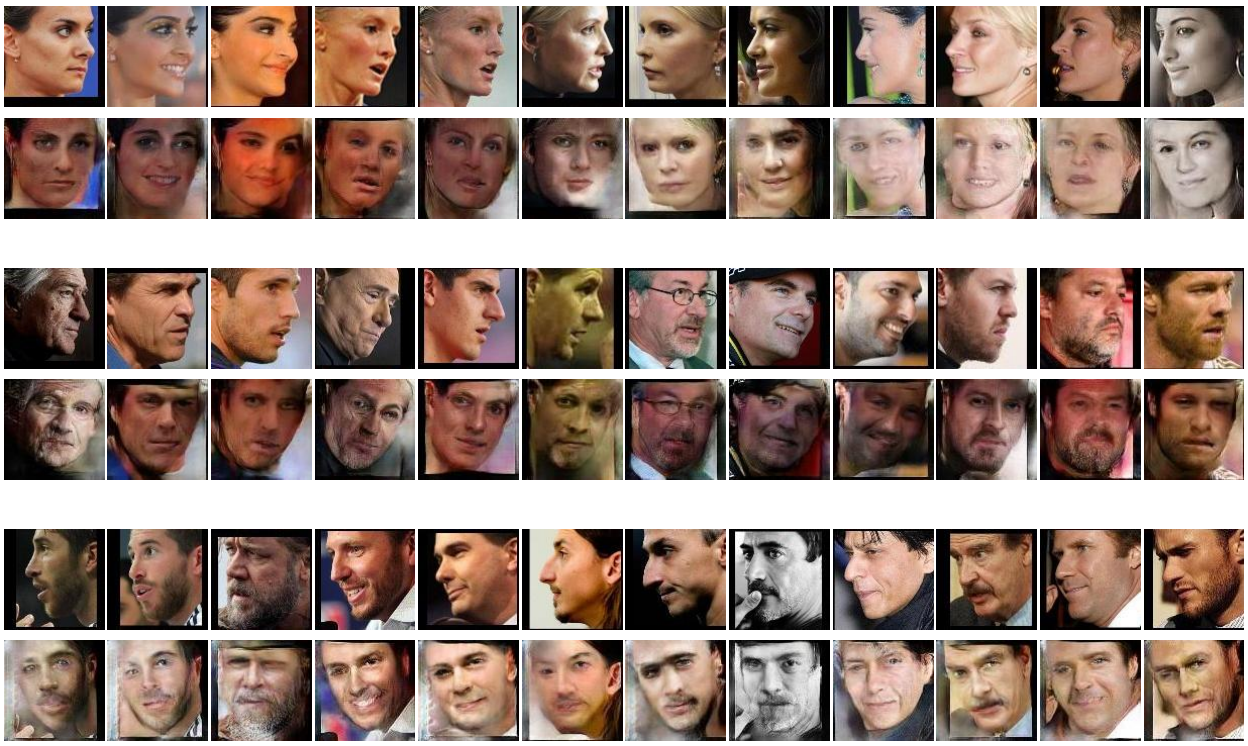


Figure 4.11: Face frontalization results on CFP. Odd rows are all profile-view inputs and even rows are the frontalized results.

Table 4.4: Quantitative results of ablation study.

removed module	–	C	D	R	G_{id}	G_{tv}	G_{sym}
performance (syn.)	**74.2**	59.2	73.4	68.5	69.3	72.9	73.1

4.4.5 Ablation Study

FF-GAN consists of four modules $\mathbb{M} = \{R, G, C, D\}$. Our generator G is the key component for image synthesis, which cannot be removed. We train three partial variants by removing each of the remaining modules, which results in $\mathbb{M}\backslash\{C\}$, $\mathbb{M}\backslash\{D\}$, and $\mathbb{M}\backslash\{R\}$. Further, we train another three variants by removing each of the three loss functions (including G_{id}, G_{tv}, G_{sym}) applied on the generated images, resulting in $\mathbb{M}\backslash\{G_{id}\}$, $\mathbb{M}\backslash\{G_{tv}\}$, and $\mathbb{M}\backslash\{G_{sym}\}$. We keep the training process and all hyper-parameters the same and explore how the performances of those models differ.

Figure 4.12 shows visual comparisons between the proposed framework and its incomplete variants. Our method is visually better than those variants, across all different poses, which suggests that each component in our model is essential for face frontalization. Without the recognizer C, it is hard to preserve identity especially on large poses. Without the discriminator D, the generated images are blurry without much high-frequency identity information. Without the reconstructor R, there are artifacts on the generated faces, which highlights the effectiveness of 3DMM in frontalization. Without the reconstruction loss G_{id}, the identity can be preserved to some extent, however the overall image quality is low, and the lighting condition is not preserved.

Table 4.4 shows the quantitative results of the ablation study models by evaluating the recognition rate of the synthetic images generated from each model. Our FF-GAN with all modules and all loss functions performs the best among all other variants, which suggests the effectiveness of each part of our framework. For example, the performance drops dramatically without the recognition engine regularization. The 3DMM module also performs a significant role in face frontalization.

Figure 4.12: Ablation study results. (a) input images. (b) $\mathbb{M}$ (ours). (c) $\mathbb{M}\backslash\{C\}$. (d)$\mathbb{M}\backslash\{D\}$. (e) $\mathbb{M}\backslash\{R\}$. (f) $\mathbb{M}\backslash\{G_{id}\}$.

4.5 Summary

In this work, we propose a 3DMM conditioned GAN framework to frontalize faces under all pose ranges including profile views. To the best of our knowledge, this is the first work to expand pose ranges to 90° in challenging large-scale datasets. The 3DMM coefficients provide an important shape and appearance prior to guide the generator to rotate faces. The recognition engine regularizes the generated image to preserve identity information. We propose new losses and carefully design the training procedure to obtain high-quality generated images. Extensive experiments consistently suggest that our frontalization algorithm may potentially boost face recognition performances and be applied for 3D face reconstruction tasks. Large-pose face frontalization is a challenging and ill-posed problem, but we believe this work has made convincing progress towards a viable solution.

Chapter 5

Feature Transfer Learning

5.1 Introduction

Face recognition is one of the ongoing success stories of the deep learning era, yielding very high accuracies on traditional datasets [62, 77, 49]. However, it remains undetermined how these results translate to practical applications, or how deep learning classifiers for fine-grained recognition must be trained to maximally exploit real-world data. While it has been established that recognition engines are data-hungry and keep improving with more volume [123], mechanisms to derive benefits from the vast diversity of real data are relatively unexplored. In particular, real-world data is long-tailed [59], with only a few samples available for most classes. In practice, effective handling of long-tail classes is also indispensable in surveillance applications where subjects may not cooperate during data collection.

It is evident that classifiers that ignore this long-tail nature of data likely imbibe hidden biases. Consider the example of the CASIA-Webface dataset [150] in Fig. 5.1(a), where about 39% of the 10K subjects have less than 20 images. A simple solution is to simply ignore the long-tail classes, as common for traditional batch construction and weight update schemes [48]. Besides reduction in the volume of data, the inherently uneven sampling leads to biases in the weight norm distribution across head and tail classes (Fig. 5.1(b,c)). Sampling tail classes at a higher frequency addresses the latter, but still leads to biased decision boundaries due to insufficient intra-class variance in tail

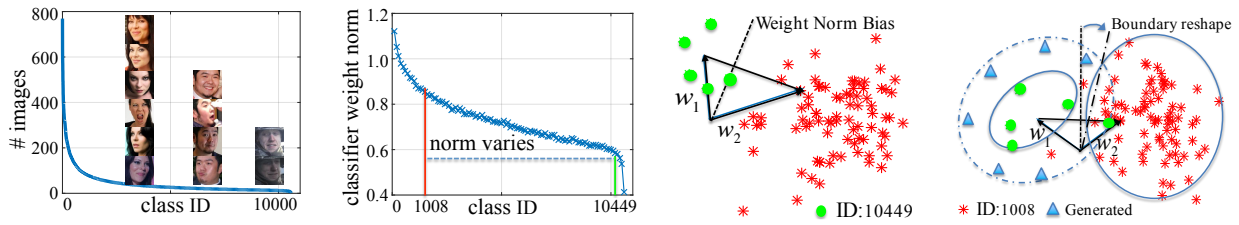


Figure 5.1: (a) The long-tail distribution of CASIA-WebFace [150]. (b) Weight norm plot of a classifier varies across classes in proportion to their volume. (c) Weight vector norm for head class ID 1008 is larger than tail class ID 10,449, causing a bias in the decision boundary (dashed line) towards ID 10,449. (d) Even after data re-sampling, the variance of ID 1008 is much larger than ID 10,449, causing decision boundary to still be biased towards the tail class. We augment the feature space of the tail classes as the dashed ellipsoid and propose improved training strategies, leading to an improved classifier.

classes (Fig. 5.1(d)).

In this work, we propose strategies for training more effective classifiers for face recognition by adapting the distribution of learned features from tail classes to mimic that of head (or regular) classes. We propose to handle long-tail classes during training by augmenting their feature space using a center-based transfer. In particular, we assume a Gaussian prior, whereby most of the variance of regular classes is captured by the top few components of a Principal Components Analysis (PCA) decomposition. By transferring the principal components from regular to long-tail classes, we encourage the variance of long-tail classes to mimic that of regular classes. Motivations for center-based transfer can also be found in recent works on the related problem of low-shot recognition [120], where the feature center is found to be a good proxy that preserves identity. Thus, restricting the transfer variance within the minimum inter-class distance limits the transfer error to be within the classifier error.

Our feature transfer overcomes the issues of imbalanced and limited training data. However, directly using the augmented data for training is sub-optimal, since the transfer might further skew the class distributions. Thus, we propose a training regimen that alternates between carefully

designed choices to solve for the feature transfer (with the goal of obtaining a less biased decision boundary) and feature learning (with the goal of learning a more discriminative representation). Further, we propose a novel metric regularization that jointly regularizes softmax feature space and weight templates, leading to empirical benefits such as reduced problems with vanishing gradients.

An approach for such feature-level transfer has also been proposed by Hariharan and Girshick [54] for 1K-class ImageNet classification [112]. But the face recognition problem is geared towards at least two orders of magnitude more classes, which leads to significant differences due to more compact decision boundaries and different nature of within-class variances. In particular, we note that the intuition of [54] to transfer semantic aspects based on relative positions in feature space is valid for ImageNet categories that vary greatly in shape and appearance, but not for face recognition. Rather, we must transfer the overall variance in feature distributions from regular to long-tail classes.

To study the empirical properties of our method, we mimic a long-tail dataset by limiting the number of samples for various proportions of classes in the MS-Celeb-1M dataset [49], while evaluating on LFW, IJB-A and the hold-out set from MS-Celeb-1M dataset. We observe that our feature transfer consistently improves upon a method that does not specifically handle long-tail classes. Moreover, we observe that adding more long-tail classes improves the overall performance of face recognition. We compare against the state-of-the-art on LFW and IJB-A benchmarks, to obtain highly competitive results that demonstrate improvement due to our feature transfer. Further, our method can be applied to challenging low-shot or one-shot scenarios, where we show competitive results on the one-shot MS-Celeb-1M challenge [48] without any tuning. Finally, we visualize our feature transfer through smooth interpolations, which demonstrate that a disentangled representation is learned that preserves identity while augmenting non-identity aspects of the feature space.

To summarize, we make the following contributions to face recognition:

- A center-based feature-level transfer algorithm to enrich the distribution of long-tailed classes, leading to diversity without sacrificing volume. It also leads to an effective disentanglement of identity and non-identity feature representation.
- A simple but effective metric regularization to enhance performances for both our method and baselines, which is also applicable to other recognition tasks.
- A two-stage alternating training scheme to achieve an unbiased classifier and retain discriminative power of the feature representation despite augmentation.
- Empirical analysis through extensive ablation studies and demonstration of benefits for face recognition in both general and one-shot settings.

5.2 Proposed Method

In Section 5.2.1, we introduce the problems caused by long-tail classes on training, such as classifier weight norm bias or intra-class variance bias, and overview challenges and solutions that will be discussed with more details in later sections. Then, we demonstrate the overall framework in Section 5.2.2 with a novel regularization. In Section 5.2.3, a center-based feature transfer method is proposed to resolve the intra-class variance bias of long-tail classes. We finally present an alternating regimen for updating the classifier with the proposed feature transfer and the feature representation to effectively train the entire system in Section 5.2.4.

5.2.1 Motivations

It is known that training deep face representations using data with long-tail distribution results in degraded performance [160]. We have similar observations in our experiments, where we train CASIA-Net [150] on CASIA-Webface [150], whose data distribution indeed shows long-tail behavior as in Fig. 5.1 (a). We further observe two atypical classifier behaviors, such as significant variations on norms of classifier weights or intra-class variances between regular and long-tail classes.

Imbalance in Classifier Weight Norm: As shown in Fig. 5.1 (b), we observe the norm of classifier weight (i.e., weight matrix of last fully connected layer) of regular classes is much larger than that of tail classes, which causes the decision boundary biases towards the tail class. This is mainly due to the fact that the weights of regular classes are more frequently updated than those of tail classes. In this regard, there exist several well-known solutions, such as data re-sampling in proportion to the volume of each class or class weights normalization [48].

Imbalance in Intra-class Variance: Unfortunately, we still observe significant imbalance after weight norm regularization via data re-sampling.[1] As an illustrative example, we randomly pick two classes, one from a regular class (ID=1008) and the other from a tail class (ID=10449). We visualize the features from two classes projected onto 2D space using t-SNE [93] in Figure 5.1(c) and those after weight norm regularization in Figure 5.1(d). Although the weights are regularized to be similar, the low intra-class variance of the tail class is not fully resolved. This causes the decision boundary to be biased, which impacts recognition performance.

We build upon this observation to posit that enlarging the intra-class variance for tail classes is the key to alleviate the impact of long-tail classes. In particular, we propose a data augmentation

[1] We found it harder to train models with weight normalization [48], nonetheless, the intra-class variance issues to which we allude would still remain.

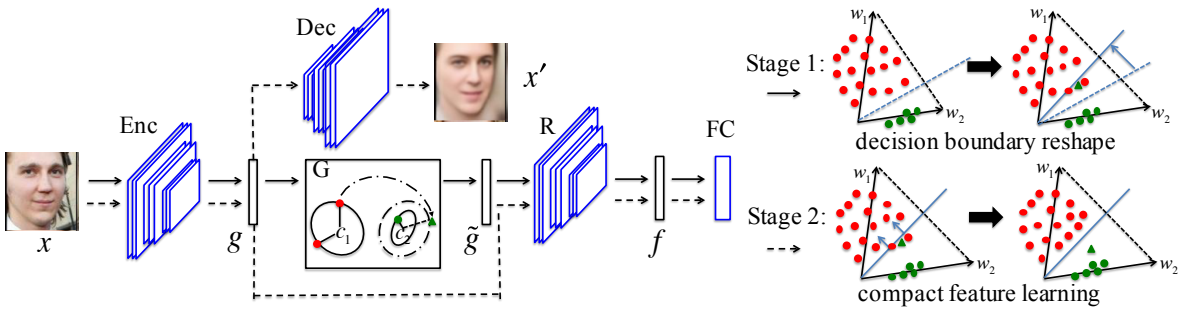


Figure 5.2: The proposed framework includes a feature extractor *Enc*, a decoder *Dec*, a feature filtering module *R*, and a fully connected layer as classifier *FC*. The proposed feature transfer module *G* generates new feature $\tilde{\mathbf{g}}$ from original feature $\mathbf{g}$. The network is trained with an alternative bi-stage strategy. At stage 1, we fix *Enc* and apply feature transfer $\mathbf{G}$ to generate new features (green triangle) that are more diverse and likely to violate decision boundary. In stage 2, we fix the rectified classifier *FC*, and update all the other models. As a result, the samples that are originally on or across the boundary are pushed towards their center (blue arrows in bottom right). Best viewed in color.

approach at the feature-level that can be used as extra positive examples for tail classes to enlarge the intra-class variance. As illustrated in Figure 5.1(d), the feature distribution augmented by these virtual positive examples helps rectify the classifier boundary, which in turn allows reshaping the feature representation.

The remainder of this section proposes specific mechanisms for regularization, feature augmentation and neural network training to realize the above intuitions.

5.2.2 Proposed Framework

Many recent successes in deep face recognition are attributable to the design of novel loss or regularization [88, 114, 106, 87, 31, 115], that reduces over-fitting to limited amount of labeled training data. In contrast, our method focuses on recovering the missing samples of tail classes by transferring knowledge from regular classes to enlarge intra-class variance. At first glance, our goal of diversifying features of tail classes appears to contradict the general premise of deep learning frameworks, which is to learn compact and discriminative features. However, we argue

that it is more advantageous to learn intra-class variance of tail classes for generalization, that is, adapting to unseen examples. To achieve this, we enlarge the intra-class variance of tail classes at a lower layer, which we call a rich feature layer [50], while subsequent filtering layers learn a compact representation with the softmax loss. Next, we define the training objectives of our proposed framework.

As illustrated in Figure 5.2, the proposed face recognition system is composed of several components, such as an encoder, decoder, feature transfer module followed by filtering and classifier layers, as well as multiple training losses, such as image reconstruction loss, or classification loss. An encoder *Enc* computes rich feature $\mathbf{g} = Enc(\mathbf{x}) \in \mathbb{R}^{320}$ of an input image $\mathbf{x} \in \mathbb{R}^{100\times100}$ and reconstruct an input image with a decoder *Dec*, i.e., $\mathbf{x}' = Dec(\mathbf{g}) = Dec(Enc(\mathbf{x})) \in \mathbb{R}^{100\times100}$. This pathway is trained with the following pixel-wise reconstruction loss:

$$\mathscr{L}_{recon} = \|\mathbf{x}' - \mathbf{x}\|_2^2 \tag{5.1}$$

The reconstruction loss allows $\mathbf{g}$ to contain diverse attributes besides identity, such as pose and expression that are to be transferred from regular classes to tail classes. A feature transfer module G transfers the variance computed from regular classes and generates a new feature $\tilde{\mathbf{g}} = G(\mathbf{g}) \in \mathbb{R}^{320}$ from tail classes, as described in the next section. Then, a filtering network R is applied to generate identity-related features $\mathbf{f} = R(\mathbf{g}) \in \mathbb{R}^{320}$ that are fed to a classifier layer FC with weight matrix $[\mathbf{w_j}] \in \mathbb{R}^{N_c\times320}$. This pathway optimizes the softmax loss:

$$\mathscr{L}_{sfmx} = -\log \frac{\exp(\mathbf{w}_i^T\mathbf{f})}{\sum_j^{N_c}\exp(\mathbf{w}_j^T\mathbf{f})}, \tag{5.2}$$

where i is the ground truth label of $\mathbf{f}$.

We note that the softmax loss is *scale-dependent*, that is, the loss can be made arbitrarily small by scaling the norm of the weights $\mathbf{w}_j$ or feature $\mathbf{f}$. Typical solutions to prevent the problem are to either regularize the norm of weights[2] or features [106], or to normalize them [48, 138]. However, we argue that these are too stringent since they penalize norms of individual weights and feature without considering their compatibility. Instead, we propose to directly regularize the norm of exponent of softmax loss as follows:

$$\mathscr{L}_{reg} = \|\mathbf{W}^T\mathbf{f}\|_2^2 \tag{5.3}$$

We term our proposed regularization a metric L_2 or m-L_2. As we will discuss in Section 5.3.2, joint regularization of weights and features through the magnitude of their inner product works better in practice than individual regularization.

Finally, we formulate the overall training loss as Equation 5.4, with the regularization coefficients set to α_{sfmx}=α_{recon}=1, and α_{reg}=0.25 unless otherwise stated:

$$\mathscr{L} = \alpha_{sfmx}\mathscr{L}_{sfmx} + \alpha_{recon}\mathscr{L}_{recon} + \alpha_{reg}\mathscr{L}_{reg}. \tag{5.4}$$

5.2.3 Long-Tail Class Feature Transfer

Following previous face recognition approaches, such as joint Bayesian face models [18, 17], we assume that rich features $\mathbf{g}_{ik}$ from class i lies in Gaussian distribution with the class-specific mean $\mathbf{c}_i$ and the covariance matrix Σ_i. To transfer intra-class variance from regular to long-tail classes, we assume the covariance matrices are shared across all classes, $\Sigma_i = \Sigma$. Under this assumption, the

[2] `http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression\#Weight_Decay`

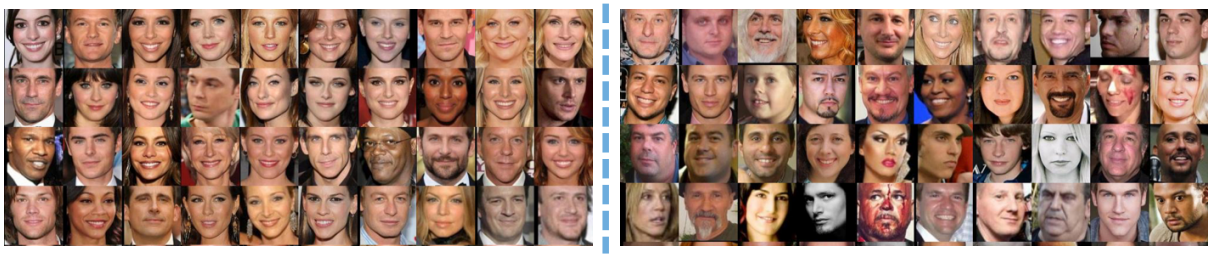


Figure 5.3: Visualization of samples closest to the feature center. (Left) We find that near-frontal close-to-neutral faces are the nearest neighbors of the feature center for regular classes. (Right) Faces closest to center are from classes with least samples, which still contain pose and expression variance, as tail classes may severely lack neutral samples. Features are extracted by VGGFace [101] and samples are from CASIA-WebFace [150].

mean, or a class center, is simply estimated as an arithmetic average of all features from the same class. As shown in the left of Figure 5.3, the center representation for regular classes is identity-specific while removing irrelevant factors of variation such as pose, expression or illumination. However, as in the right of Figure 5.3, due to lack of training examples, the center estimate of long-tail classes is not accurate and often biased towards certain identity-irrelevant factors, such as pose, which we find dominant in practice. To improve the quality of center estimate for long-tail classes, we discard examples with extreme pose variations. Furthermore, we consider averaging features from both the original and horizontally flipped images. With $\bar{\mathbf{g}}_{ik} \in \mathbb{R}^{320}$ a rich feature extracted from the flipped image, the feature center is estimated as follows:

$$\mathbf{c}_i = \frac{1}{2|\Omega_i|} \sum_{k \in \Omega_i} (\mathbf{g}_{ik} + \bar{\mathbf{g}}_{ik}), \qquad \Omega_i = \{j \,|\, \|p_{ik} - \bar{p}_{ik}\|_2 \leq \tau\}, \tag{5.5}$$

where p_{ik} and $\bar{p}_{ik}$ are the pose codes for $\bar{\mathbf{g}}_{ik}$ and $\mathbf{g}_{ik}$, respectively. Ω_i includes indices for examples with yaw angle less than a threshold τ.

Next, we transfer the variance estimated from the regular classes to long-tail classes. In theory, one can draw feature samples of long-tail classes by adding a noise vector $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma)$. However, the direction of noise vectors might be too random when sampled from the distribution and does not

reflect the true factors of variation found in the regular classes. Instead, we transfer the intra-class variance evaluated from individual samples of regular classes. To further remove identity-related component in the variance, we filter them using PCA basis $\mathbf{Q} \in \mathbb{R}^{320\times150}$ [142] achieved from intra-class variances of all regular classes. We take the top 150 Eigen vectors as preserving 95% energy. Our center-based feature transfer is achieved using:

$$\tilde{\mathbf{g}}_{ik} = \mathbf{c}_i + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_{jk} - \mathbf{c}_j), \tag{5.6}$$

where $\mathbf{g}_{jk}$ and $\mathbf{c}_j$ are a feature sample and center of a regular class j, $\mathbf{c}_i$ is the feature center of a long-tail class i and $\tilde{\mathbf{g}}_{ik}$ is the transferred feature for class i. Here, $\tilde{\mathbf{g}}_{ik}$ preserves the same identity as $\mathbf{c}_i$, with similar intra-class variance as $\mathbf{g}_{jk}$. By sufficiently sampling $\mathbf{g}_{jk}$ across different regular classes, we expect to obtain an enriched distribution of the long-tail class i, which consists of both the original observed features $\mathbf{g}_{ik}$ and the transferred features $\tilde{\mathbf{g}}_{ik}$.

5.2.4 Alternating Training Strategy

Given a training set of regular and long-tail classes $\mathbb{D} = \{\mathbb{D}_{reg}, \mathbb{D}_{lt}\}$, we first train all modules $\mathbb{M} = \{Enc, Dec, R, FC\}$ using Equation 5.4 without any feature transfer. Then, we alternatively train a classifier until convergence with decision boundary reshaping using our proposed feature transfer and a feature representation with boundary-corrected classifier. The overview of our two-stage alternating training process is illustrated in Algorithm 5.1 and 5.2. We describe in more details of each training stage below.

Stage 1: Decision Boundary Reshape. In this stage, we update R and FC while fixing other modules using variance transferred features from regular to long-tail classes to enlarge the intra-

Algorithm 5.1: Alternating training scheme for feature transfer learning.

Stage 0: model pre-training
  train $\mathbb{M}$ with dataset $\mathbb{D}$ using Eqn. 5.4
Stage 1: decision boundary reshape
  Fix *Enc* and *Dec*, train *R* and *FC*
  $[\mathbf{C}, \mathbf{Q}, \mathbf{h}]$ = ***UpdateStats***()
  Init $G(\mathbf{C}, \mathbf{Q})$
  for $i = 1, \ldots, N_{iter}$ **do**
    train 1st batch from $\mathbf{h}$: $\{\mathbf{x}^r, \mathbf{y}^r\}$
    train 2nd batch from $\mathbb{D}_{lt}$: $\{\mathbf{x}^t, \mathbf{y}^t\}$
    $\tilde{\mathbf{g}}^t$ = ***Transfer***($\mathbf{x}^r$, $\mathbf{y}^r$, $\mathbf{y}^t$)
    train 3rd batch: $\{\tilde{\mathbf{g}}^t, \mathbf{y}^t\}$
Stage 2: compact feature learning
  Fix *FC*, train *Enc*, *Dec*, and *R*
  for $i = 1, \ldots, N_{iter}$ **do**
    random samples from $\mathbb{D}$: $\{\mathbf{x}, \mathbf{y}\}$
    train $\{\mathbf{x}, \mathbf{y}\}$ using Eqn. 5.4
alternate stage 1 and 2 until convergence

class variance of long-tail classes, thus, reshape the decision boundary. We first update the statistics including the feature centers $\mathbf{C}$, PCA basis $\mathbf{Q}$ and an index list $\mathbf{h}$ of hard samples that are with the distance from the center more than the average distance for each regular class. The PCA basis $\mathbf{Q}$ is achieved by decomposing the covariance matrix $\mathbf{V}$ computed with the samples from regular classes $\mathbb{D}_{reg}$. Three batches are used for training in each iteration: a regular batch sampled from hard index list $\mathbf{h}$: $\{\mathbf{g}^r, \mathbf{y}^r\}$, a long-tail batch sampled from long-tail classes $\{\mathbf{g}^t, \mathbf{y}^t\}$, and a transferred batch $\{\tilde{\mathbf{g}}^t, \mathbf{y}^t\}$ by transferring the variances from regular batch to long-tail batch.

Stage 2: Compact Feature Learning. In this stage, we train *Enc*, *Dec* as well as *R* using normal batches $\{\mathbf{x}, \mathbf{y}\}$ from regular and long-tail classes using Equation 5.4 without transferred batch. We keep *FC* fixed since it is already trained well from the previous stage with decision boundary corrected using feature transfer. The gradient directly back-propagates to *R* and *Enc* for more compact representation, which decreases violation of class boundaries.

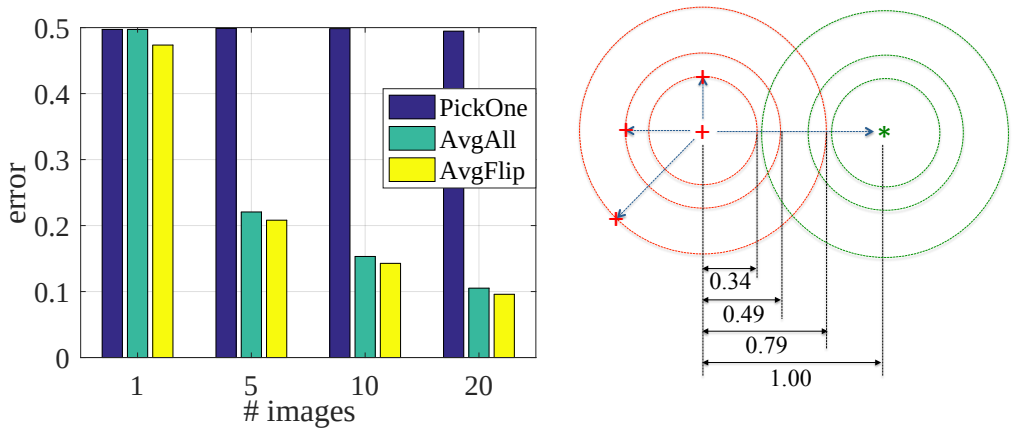
Algorithm 5.2: Functions that are called in Algorithm 5.1.

Function $[\mathbf{C}, \mathbf{Q}, \mathbf{h}]$ = ***UpdateStats***()
Init $\mathbf{C} = []$, $\mathbf{V} = []$, $\mathbf{h} = []$
for $i = 1, \ldots, N_c$ **do**
  $\mathbf{g}_i = Enc(\mathbf{x}_i)$, $\bar{\mathbf{g}}_i = Enc(\bar{\mathbf{x}}_i)$
  $\mathbf{c}_i = \frac{1}{2|\Omega_i|} \sum_{j \in \Omega_i} (\mathbf{g}_{ij} + \bar{\mathbf{g}}_{ij})$
  $\mathbf{C}$.append($\mathbf{c}_i$)
  if *i in* $\mathbb{D}_{reg}$ **then**
    $d = \frac{1}{m_i} \sum_j ||\mathbf{g}_{ij} - \mathbf{c}_i||_2$
    for $j = 1, \ldots, m_i$ **do**
      $\mathbf{V}$.append($\mathbf{g}_{ij} - \mathbf{c}_i$)
      if $||\mathbf{g}_{ij} - \mathbf{c}_i||_2 > d$ **then**
        $\mathbf{h}$.append([i,j])
$\mathbf{Q} = \text{PCA}(\mathbf{V})$
Function $\tilde{\mathbf{g}}^t$ = ***Transfer***($\mathbf{x}^r$, $\mathbf{y}^r$, $\mathbf{y}^t$)
$\mathbf{g}^r = Enc(\mathbf{x}^r)$
for $k = 1, \ldots, N_b$ **do**
  $\mathbf{c}_i = \mathbf{C}(\mathbf{y}^r_k, :)$, $\mathbf{c}_j = \mathbf{C}(\mathbf{y}^t_k, :)$
  $\tilde{\mathbf{g}}^t_k = \mathbf{c}_i + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}^r_k - \mathbf{c}_j)$

5.3 Experimental Results

We train our models on MS-Celeb-1M dataset [49], which consists of 10M images from around 100K celebrities. Due to label noise, we adopt a cleaned version from [143] and further remove the subjects overlapped with LFW and IJB-A, which results in 4.8M images of 76.5K classes for training. A class with no more than 20 images is considered as a long-tail class, following [160].

For implementation, we apply encoder-decoder structure similar to [132] and ResNet-54 for *Enc* in Section 5.3.5. Model *R* consists of a *FC* layer, two full convolution layers, two convolution layers and another *FC* layer to achieve $\mathbf{f} \in \mathbb{R}^{320 \times 1}$. More detail is referred to supplementary material. Adam solver with learning rate $2e^{-4}$ is used in stage 0. Learning rate $1e^{-5}$ is used in stage 1 and 2, which alternated for every 5K iterations.



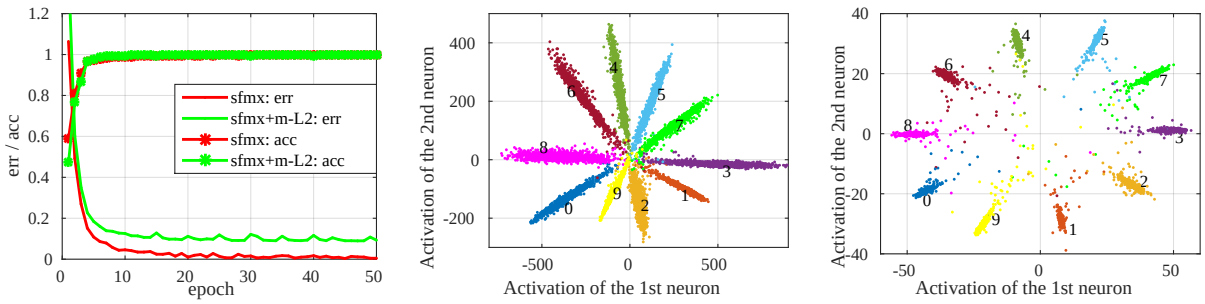
(a) (b)

Figure 5.4: (a) Center estimation error comparison. (b) Two classes with intra-class and inter-class variance illustrated. Circles from small to large show minimum, mean and maximum distance from intra-class samples to the center. Distances are averaged across 1K classes.

5.3.1 Feature Center Estimation

Feature center estimation is a key step for feature transfer. To evaluate center estimation for tail classes, 1K regular classes are selected from MS-Celeb-1M and features are extracted using a pretrained recognition model. We randomly select a subset of 1, 5, 10, 20 images to mimic a long-tail class. Three methods are compared: (1) “PickOne”, randomly pick one sample as center. (2) “AvgAll”, average feature of all images. (3) “AvgFlip”, proposed method in Equation 5.5. Error is the difference between the center of full set (ground truth) and the subset. Intra-class and inter-class variance are provided as reference. All errors are normalized by the inter-class variance.

Results in Figure 5.4 show that our “AvgFlip" achieves clear smaller error. When compared to the intra-class variance, the error is fairly smaller, which convince that our center estimate is accurate to support the feature transfer.



(a) (b) (c)

Figure 5.5: Toy example on MNIST to show the effectiveness of our m-L_2 regularization. (a) the training loss/accuracy comparison. (b) feature distribution on test set for the model trained without m-L_2 regularization. (c) feature distribution with m-L_2 regularization.

5.3.2 Effects of m-L2 Regularization

To study the effects of the proposed m-L_2 regularization, we show a toy example on the MNIST dataset [79]. We use LeNet++ network [141] to learn a 2D feature for better visualization. Two models are trained: one with softmax loss only; the other with softmax loss and m-L_2 regularization ($\alpha_{reg} = 0.001$).

m-L_2 regularization has several advantages. (1) m-L_2 effectively avoids over-fitting. In Figure 5.5, $softmax$ training shows over-fitting as training error goes to 0 whereas with m-L_2 training error stays small but not 0. (2) m-L_2 enforces a more balanced feature distribution. Figure 5.5 (c) shows a more balanced angular distribution than Figure 5.5 (b). The performance with m-L_2 improves $softmax$ from 99.06% to 99.35%. We believe m-L_2 is a simple yet powerful general regularization that can be easily adapted to other recognition problems.

5.3.3 Ablation Study

We study two factors to analyze the long-tail training: (1) the ratio of the portion of regular classes vs. the portion of long-tail classes; (2) the number of images per long-tail class. We use discrete

Table 5.1: Results on the controlled experiments by varying the ratio between regular and long-tail classes in the training sets.

Test →		LFW		IJB-A: Verif.		IJB-A: Identif.		MS1M: NN	
Train↓	Method↓	**g**	**f**	FAR@.01	@.001	Rank-1	Rank-5	Reg.	LT
10K0K	sfmx	97.15	97.45	69.39	33.04	81.63	90.35	87.17	82.47
	sfmx+m-L_2	97.00	97.88	73.00	44.78	83.77	91.49	90.21	84.68
10K10K	sfmx	–	97.85	72.96	49.22	82.38	90.46	85.87	85.25
	sfmx+m-L_2	97.08	97.85	74.07	46.27	83.70	91.74	89.48	84.10
	Ours	96.72	98.33	80.25	54.95	85.88	92.83	92.27	88.16
10K30K	sfmx	–	97.80	74.03	47.93	83.04	91.25	86.14	85.47
	sfmx+m-L_2	97.13	98.08	76.92	47.17	84.81	91.93	90.60	86.40
	Ours	96.87	98.42	81.80	61.04	86.08	92.62	91.76	88.72
10K50K	sfmx	–	97.93	72.87	49.04	82.40	91.15	85.28	84.21
	sfmx+m-L_2	97.32	98.10	78.52	53.44	84.95	92.17	90.24	87.11
	Ours	96.95	98.48	82.60	62.60	86.53	93.08	92.08	89.36
60K0K	sfmx	97.52	98.30	82.75	62.33	87.11	93.78	90.43	89.54
	sfmx+m-$L2$	**97.90**	**98.85**	**86.38**	**74.44**	**89.34**	**94.65**	**93.68**	**93.46**

approximation to mimic the real regular vs. long-tail class distribution and the continuous distribution of number of samples per tail class. Our main focus is to analyze the long-tail distribution impact on recognition thus assume discrete setting for simplicity.

Regular/Long-Tail Class Ratio: we use 60K regular classes with most number of images from MS-Celeb-1M. The top 10K classes are selected as regular classes which are shared among all training sets. We regard the 10K and 60K sets to serve as the lower and upper bounds. Among the rest 50K classes sorted by number of images, we select the first 10K, 30K and 50K and randomly pick 5 images per class. In this way, we form the training set of 10K10K, 10K30K, and 10K50K, of which the first 10K are regular and the last 10K or 30K or 50K are called faked long-tail classes. A hold-out testing set is formed by selecting 5 images from each of the shared 10K regular classes and 10K tail classes, resulting in 100K testing images.

The evaluation on the hold out test set from MS-Celeb-1M is to mimic the low-shot learning

scenario, where we use the feature center from the training images as the gallery and nearest neighbor (NN) for face matching. The rank-1 accuracy for both regular and long-tail classes are reported. We also evaluate the general face recognition performance on LFW and IJB-A. The results are shown in Table 5.1 and we draw the following observations.

- The feature space **g** is less discriminative than the feature space **f**, which validates our assumption that **g** is rich in intra-class variance for feature transfer while **f** is more discriminative for face recognition.

- The proposed m-L_2 regularization boosts the performance with a large margin over the baseline softmax loss.

- The proposed transfer method consistently improves over sfmx and sfmx+m-L_2 with significant margins, and largely close the gap from 10K0K to 60K0K.

- Our method is more beneficial when more long-tail classes are added to training as more long-tail classes lead to better face recognition performance.

Number of Images per Long-Tail Class: we vary $n = 1, 5, 10, 20$ under setting 10K30K. Table 5.2 reveals that more images in long-tail classes leads to better results, due to better center estimation. Consistent with Table 5.1, the proposed algorithm significantly improves performance on low-shot setting of MS-Celeb-1M and general face recognition on LFW and IJB-A. On 10K30K ($n = 5$) setting, we look into the FC classifier performance, 93.59% and 2.04% for regular and long-tail respectively. Whereas our method achieves 96.26% and 81.89% accordingly, which suggests our method's effectiveness in correcting classifier bias.

Table 5.2: Results of the controlled experiments by varying the number of images for each long-tail class in the training sets.

Test →		LFW	IJB-A: Verif.		IJB-A: Identif.		MS1M:NN	
Train ↓	Method↓	**f**	FAR@.01	@.001	Rank-1	Rank-5	Reg.	LT
10K30K ($n=1$)	sfmx	97.82	72.03	43.56	82.51	91.01	87.35	86.94
	sfmx+m-L_2	97.93	74.22	47.79	83.94	91.52	90.47	84.85
	Ours	98.28	78.65	51.15	85.82	92.23	92.65	88.99
10K30K ($n=5$)	sfmx	97.80	74.03	47.93	83.04	91.25	86.14	85.47
	sfmx+m-L_2	98.08	76.92	47.17	84.81	91.93	90.60	86.40
	Ours	98.42	81.80	61.04	86.08	92.62	91.76	88.72
10K30K ($n=10$)	sfmx	97.98	75.67	52.48	83.41	91.34	86.04	85.93
	sfmx+m-L_2	98.38	80.11	56.51	86.00	93.11	90.83	88.77
	Ours	98.60	84.07	64.73	87.55	93.72	92.89	90.89
10K30K ($n=20$)	sfmx	98.08	76.36	54.14	83.68	91.77	86.42	86.76
	sfmx+m-L_2	98.58	80.61	59.75	86.34	93.36	91.40	90.05
	Ours	**98.83**	**85.27**	**67.19**	**88.42**	**94.14**	**93.38**	**92.26**

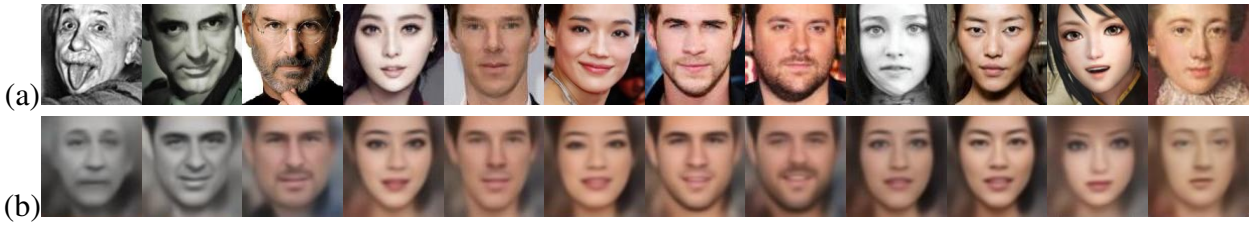


Figure 5.6: Center visualization: (a) one sample image from the selected class; (b) the decoded image from the feature center.

5.3.4 One-Shot Face Recognition

While our method has only tangential relation to one-shot recognition, we evaluate on the MS1M one-shot challenge as an illustration [48]. In this setting, the training data consists of a base set with 20K classes each with $50 \sim 100$ images and a novel set of 1K classes each with only 1 image. The test set consists of 1 image for each base class and 5 images for each novel class. The main purpose is to evaluate the recognition performance on the novel classes while monitoring the performance on the base classes.

Table 5.3: Comparison on one-shot MS-Celeb-1M challenge. Results on the base classes are reported as rank-1 accuracy and on novel classes are reported as Coverage@Precision = 0.99.

Method	External Data	Models	Base	Novel
MCSM [148]	YES	3	–	61.0
Cheng et al. [25]	YES	4	99.74	**100**
Choe et al. [27]	NO	1	$\geq$ 95.00	11.17
UP [48]	NO	1	99.80	77.48
Hybrid [144]	NO	2	99.58	**92.64**
DM [119]	NO	1	–	73.86
Ours	NO	1	99.21	92.60

Table 5.4: Face recognition on LFW and IJB-A. "MP" represents media pooling and "TA" represents template adaptation. The best and second-best results are highlighted.

Test $\rightarrow$	LFW
MTL [153]	98.27
L-Softmax [88]	98.71
VGG Face [101]	98.95
DeepID2 [125]	99.15
NormFace [138]	99.19
CenterLoss [141]	99.28
SphereFace [87]	99.42
RangeLoss [160]	**99.53**
FaceNet [115]	**99.63**
sfmx	98.60
sfmx + L_2	98.53
sfmx + m-L_2	99.18
Ours	99.37

Test $\rightarrow$	IJB-A: Verif.		IJB-A: Identif.		
Method $\downarrow$	FAR@.01	@.001	Rank-1	-5	-10
PAMs [95]	82.6	65.2	84.0	92.5	94.6
DR-GAN [132]	83.1	69.9	90.1	95.3	–
FF-GAN [154]	85.2	66.3	90.2	95.4	–
TA [31]	**93.9**	–	92.8	–	**98.6**
TPE [114]	90.0	81.3	86.3	93.2	97.7
NAN [149]	**94.1**	**88.1**	**95.8**	**98.0**	**98.6**
sfmx	86.5	71.0	88.7	94.5	96.1
sfmx + L_2	84.5	68.1	88.6	94.9	96.4
sfmx + m-L_2	90.6	80.5	92.3	96.3	97.2
Ours	91.0	81.0	92.7	96.4	97.4
Ours + MP	92.1	83.8	93.3	**96.7**	**97.7**
Ours + MP + TA	93.1	**87.3**	**93.9**	96.6	97.5

As shown in Table 5.3, we achieve 95.48% rank-1 accuracy with a single model and single crop testing. We use the output from softmax layer as the confidence score and achieve 92.60% coverage at precision of 0.99. Note that the best models [144] and [25] use model ensembling with different crops for testing. Compared to similar setting methods [48, 27], we achieve competitive performance on the base classes and much better results on the novel classes.

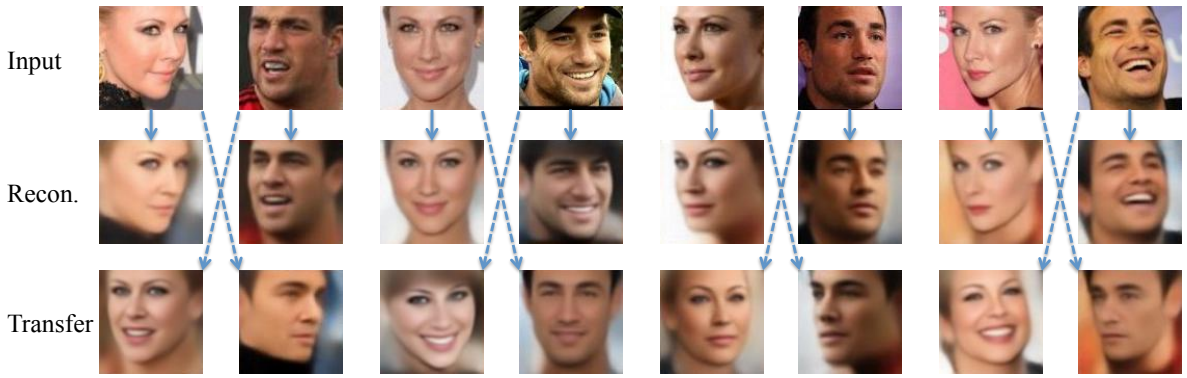


Figure 5.7: Feature transfer visualization between two classes for every two columns. The first row are the input, in which odd column denotes class 1: $\mathbf{x}_1$ and the even column denotes class 2: $\mathbf{x}_2$. The second row are the reconstructed images $\mathbf{x}'_1$ and $\mathbf{x}'_2$. In the third row, odd column image is the decoded image of the transferred feature from class 1 to class 2 and even column image is the decoded image of the transferred feature from class 2 to class 1. It is clear that the transferred features share the same identity as the target class while obtain the source image's non-identity variance including pose, expression, illumination, and etc.

5.3.5 Large-Scale Face Recognition

In this section, we train our model on the full MS-celeb-1M dataset and evaluate on LFW and IJB-A. The cleaned dataset includes 76.5K classes, of which 9.5K classes consist of less than 20 images. We use a ResNet-54 structure for *Enc*. As shown in Table 5.4, the deeper network structure with our proposed m-L_2 regularization already provides good results. Our feature transfer learning further improves the performance significantly. On LFW, our performance is among the state-of-the-art. On IJB-A, our method significantly outperforms most of the methods except "NAN". While "NAN" is designed with attention or aggregation model to specifically incorporate temporal information, our method is geared towards still image recognition with long-tail classes.

5.3.6 Qualitative Results

We apply decoder *Dec* in our framework for feature visualization. It is well known that the skip link between an encoder and decoder can improve visual quality [154]. However, we do not apply

it in order to encourage the feature $\mathbf{g}$ to incorporate the intra-class variance other than from the skip link.

Center Visualization Given a class with multiple samples, we compute a feature center, on which the *Dec* is applied to generate a center face. From Figure 5.6, we confirm the observation that the center is mostly an identity-preserved frontal neutral face. It also applies to portrait and cartoon figures.

Feature Transfer The transferred feature is visualized using the *Dec*. Let $\mathbf{x}_{1,2}$, $\mathbf{x}'_{1,2}$, $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ denote the input images, reconstructed images, encoded features and feature centers of two classes, respectively. We transfer feature from class 1 to class 2 by: $\mathbf{g}_{12} = \mathbf{c}_2 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_1 - \mathbf{c}_1)$, and visualize the decoded images. We also transfer from class 2 to class 1 and visualize the decoded images. Figure 5.7 shows the examples of feature transfer between two classes. The transferred images preserve the target class identity while retaining intra-class variance of the source in terms of pose, expression and lighting, which shows that our feature transfer is effective at enlarging the intra-class variance.

Feature Interpolation The interpolation between two facial representations shows the appearance transition from one to the other [105, 132]. Let $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ denote the encoded features and the feature centers of two samples. Previous work generates a new representation as $\mathbf{g} = \mathbf{g}_1 + \alpha(\mathbf{g}_2 - \mathbf{g}_1)$ where identity and non-identity changes are mixed together. In our work, we can generate a smooth transition of non-identity change as $\mathbf{g} = \mathbf{c}_1 + \alpha\mathbf{Q}\mathbf{Q}^T(\mathbf{g}_2 - \mathbf{c}_2)$ and identity change as $\mathbf{g} = \mathbf{g}_1 + \alpha(\mathbf{c}_2 - \mathbf{c}_1)$. Figure 5.8 shows an interpolation example of a female with left pose and a male with right pose, where the illumination also changes significantly. Traditional interpolation generates undesirable artifacts. However, our method shows smooth transitions, which verifies that the proposed model is effective at disentangling identity and non-identity information.

5.4 Summary

In this work, we propose a novel feature transfer approach for deep face recognition that exploits the long-tailed nature of training data. We observe that generic approaches to deep face recognition encounter classifier bias problems due to imbalanced distribution of training data across identities. In particular, uniform sampling of both regular and long-tail classes leads to biased classifier weights, since the frequency of updating them for long-tail classes is much lower. By applying the proposed feature transfer approach, we enrich the feature space of the tail classes, while retaining identity. Utilizing the generated data, our alternating feature learning method rectifies the classifier and learns more compact feature representations. Our proposed m-L_2 regularization demonstrates consistent advantages which can boost performance across different recognition tasks. The disentangled nature of the augmented feature space is visualized through smooth interpolations. Experiments consistently show that our method can learn better representations to improve the performance on regular, long-tail and unseen classes. While this work focuses on face recognition, our future work will also derive advantages from the proposed feature transfer for other recognition applications, such as long-tail natural species.

Figure 5.8: Transition from top-left image to top-right image via feature interpolation. For each example, first row shows traditional feature interpolation; second row shows our transition of non-identity variance; third row shows our transition of identity variance.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Face recognition is an important research topic because it has many real-world applications in surveillance, law enforcement, commercial systems, and etc. With deep neural networks, the performance on benchmark datasets improve dramatically, which contributes to the deployment of face recognition systems in our daily life. For example, face starts to replace fingerprint for unlocking personal devices. Companies are using a face recognition system to control building access. Railway stations provide self-serviced check-in that compares a customer's face with his or her ID photo to make the transit process more efficient. These applications have driven the face recognition research for several decades.

Throughout this dissertation, we have presented three different approaches from the perspective of representation learning and image synthesis for deep face recognition. Representation learning is the essential component in designing a face recognition method. It is challenging but also promising with the development of novel network structures and loss functions. On the other hand, image synthesis has the advantage of providing visual appealing results for better understanding. These two methods are often not independent from each other. A good representation is very important for identity-preserved face image synthesis. Meanwhile, synthesized images can provide complementary features for representation learning. We have made some efforts from both

perspectives to advance state-of-the-art deep face recognition in this dissertation.

6.2 Future Work

While the recognition performance on benchmark face databases improves dramatically [115, 33, 139], unconstrained face recognition is still not a solved problem as it can encounter a lot of failure cases in real-world deployment. Such failure cases include faces captured with heavy occlusions (hair, eyeglasses, etc) or under bad lighting conditions. One application with increasing interest is surveillance face recognition, which is challenging as the face images are often with very low quality. Improving the recognition performance of low-quality face images is important to facilitate surveillance face recognition. Moreover, the generalization ability of the learnt representation is usually very poor when the test data distribution is different from the training data. This is due to the limited understanding of the representation learnt in a DNN framework. We are interested in these two aspects for future work.

Surveillance Face Recognition Face images captured with surveillance cameras are different from those celebrity face images collected from the internet. Surveillance Face Recognition (SFR) is difficult mainly due to the absence of surveillance data. Fortunately, it is attracting more attentions lately. The UCCS dataset [47] is released for unconstrained face detection and open-set face recognition from surveillance videos. Cheng et al. [26] introduce the SFR challenge, where state-of-the-art recognition algorithms are still far from being satisfactory in the surveillance scenario. Apparently, more research efforts are needed to tackle this problem.

Similar to current face recognition techniques, representation learning, and image synthesis are the two main directions to explore SFR. Representation learning-based methods [140] are less studied compare to synthesis-based methods of face super-resolution, where facial priors like land-

mark and attributes are used for image restoration [14, 24, 155]. Although face super-resolution is a promising direction, the current experimental setups are based on down-sampled low-resolution images from the original high-resolution face images, which is quite different compared to the real low-resolution / low-quality images. We will explore how to better combine representation learning and image synthesis for SFR.

Feature Interpretation Unlike traditional features (LBP, HOG, SIFT, etc) where the feature extraction process is well-defined, deep features, though being more discriminative, are less interpretable. While a tremendous number of work focuses on the design of the representation learning methods, only limited work is introduced to understand the learnt representation. Gong et al. [42, 43] are the first to study the capacity and the intrinsic dimension of a face representation. It provides insights that state-of-the-art face representation (the 128-d FaceNet features) are with high capacity and low intrinsic dimension. This work raises the question of how many dimensions are really needed for face recognition, which is relatively unexplored in the face recognition community.

Besides studying the dimension of the representation, the meaning of the representation, or the logic inside a CNN framfework is also studied in [159, 152]. Zhang et al. [159] introduce interpretable CNN that automatically assigns each filter in the convolution layer with an object part during training. Such a formulation can encode more meaningful knowledge into the high convolution layers at the price of decreased classification accuracy. Yin et al. [152] propose interpretable face recognition to encourage the diversity of different filters and the learnt representation, which is shown to improve the performance. However, there is still a gap compared to state-of-the-art non-interpretable face recognition methods. How to learn interpretable representation that can achieve competitive performance is an interesting problem to study in the future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. Multi-task CNN model for attribute prediction. *TMM*, 2015.

[2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (12):2037–2041, 2006.

[3] S. Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

[4] E. N. Arcoverde Neto, R. M. Duarte, R. M. Barreto, J. P. Magalhães, C. Bastos, T. I. Ren, and G. D. Cavalcanti. Enhanced real-time head pose estimation system for mobile device. *Integrated Computer-Aided Engineering*, 2014.

[5] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.

[6] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3D pose normalization. In *ICCV*, 2011.

[7] A. Asthana, C. Sanderson, T. D. Gedeon, R. Goecke, et al. Learning-based face synthesis for pose-robust recognition from single image. In *BMVC*, 2009.

[8] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Towards open-set identity preserving face synthesis. In *CVPR*, 2018.

[9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Technical report, Yale University New Haven United States, 1997.

[10] L. Best-Rowden and A. K. Jain. Longitudinal study of automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[11] S. Biswas, G. Aggarwal, P. J. Flynn, and K. W. Bowyer. Pose-robust recognition of low-resolution face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[12] V. Blanz and T. Vetter. A morphable model for the synthesis of 3*d* faces. ACM Press/Addison-Wesley Publishing Co., 1999.

[13] G. Brazil, X. Yin, and X. Liu. Illuminating pedestrians via simultaneous detection and

segmentation. In *ICCV*, 2017.

[14] A. Bulat and G. Tzimiropoulos. Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. In *CVPR*, 2018.

[15] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy. Pose-robust face recognition via deep residual equivariant mapping. In *CVPR*, 2018.

[16] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018.

[17] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *ICCV*, 2013.

[18] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*, 2012.

[19] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *CVPR*, 2013.

[20] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, 2014.

[21] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep CNN features. In *WACV*, 2016.

[22] J.-C. Chen, J. Zheng, V. M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *ICIP*, 2016.

[23] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[24] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *CVPR*, 2018.

[25] Y. Cheng, J. Zhao, Z. Wang, Y. Xu, K. Jayashree, S. Shen, and J. Feng. Know you at one glance: A compact vector representation for low-shot learning. In *ICCV workshop*, 2017.

[26] Z. Cheng, X. Zhu, and S. Gong. Surveillance face recognition challenge. *arXiv preprint arXiv:1804.09691*, 2018.

[27] J. Choe, S. Park, K. Kim, J. Hyun Park, D. Kim, and H. Shim. Face generation for low-shot learning using generative adversarial networks. In *ICCV workshop*, 2017.

[28] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[29] B. Chu, S. Romdhani, and L. Chen. 3D-aided face recognition robust to expression and pose variations. In *CVPR*, 2014.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[31] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. In *FG*, 2017.

[32] J. Daugman. How iris recognition works. In *The Essential Guide to Image Processing*, pages 715–739. 2009.

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

[34] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[35] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[36] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2016.

[37] C. Ding, C. Xu, and D. Tao. Multi-task pose-invariant face recognition. *IEEE Transactions on Image Processing (TIP)*, 2015.

[38] L. El Shafey, C. McCool, R. Wallace, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[39] J. F. Fagan III. Infants' recognition of invariant features of faces. *Child Development*, 1976.

[40] C. Ferrari, G. Lisanti, S. Berretti, and A. Bimbo. Effective 3D based frontalization for unconstrained face recognition. In *ICPR*, 2016.

[41] H. Gao, H. Ekenel, and R. Stiefelhagen. Pose normalization for local appearance-based face recognition. *Advances in Biometrics*, 2009.

[42] S. Gong, V. N. Boddeti, and A. K. Jain. On the capacity of face representation. *arXiv preprint arXiv:1709.10433*, 2017.

[43] S. Gong, V. N. Boddeti, and A. K. Jain. On the intrinsic dimensionality of face representation. *arXiv preprint arXiv:1803.09672*, 2018.

[44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[45] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 2010.

[46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *NIPS*, 2017.

[47] M. Günther, P. Hu, C. Herrmann, C. H. Chan, M. Jiang, S. Yang, A. R. Dhamija, D. Ramanan, J. Beyerer, J. Kittler, et al. Unconstrained face detection and open-set face recognition challenge. In *IJCB*, 2017.

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

[49] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[50] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014.

[51] N. Hadad, L. Wolf, and M. Shahar. A two-step disentanglement method. In *CVPR*, 2018.

[52] H. Han, S. Shan, X. Chen, S. Lao, and W. Gao. Separability oriented preprocessing for illumination-insensitive face recognition. In *ECCV*, 2012.

[53] H. Han, S. Shan, L. Qing, X. Chen, and W. Gao. Lighting aware preprocessing for face recognition across varying illumination. In *ECCV*, 2010.

[54] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017.

[55] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *CVPR*, 2015.

[56] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[57] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai. Triplet-center loss for multi-view 3*d* object retrieval. In *CVPR*, 2018.

[58] E. Hjelmås and B. K. Low. Face detection: A survey. *Computer Vision and Image Under-*

standing, 83(3):236–274, 2001.

[59] G. V. Horn and P. Perona. The devial is in the tails: Fine-grained classification in the wild. In *arXiv preprint arXiv:1709.01450*, 2017.

[60] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[61] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[62] 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. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[63] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017.

[64] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[65] A. K. Jain and S. Z. Li. *Handbook of face recognition*. Springer, 2011.

[66] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

[67] H. Jiang and E. Learned-Miller. Face detection with the faster R-CNN. In *FG*, 2017.

[68] A. Jourabloo and X. Liu. Pose-invariant 3D face alignment. In *ICCV*, 2015.

[69] A. Jourabloo and X. Liu. Large-pose face alignment via CNN-based dense 3D model fitting. In *CVPR*, 2016.

[70] A. Jourabloo and X. Liu. Pose-invariant face alignment via CNN-based dense 3D model fitting. *International Journal of Computer Vision*, 124(2):187–203, 2017.

[71] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (SPAE) for face recognition across poses. In *CVPR*, 2014.

[72] M. Kan, S. Shan, and X. Chen. Multi-view deep network for cross-view classification. In *CVPR*, 2016.

[73] N. Kanwisher, J. McDermott, and M. M. Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17(11):4302–4311, 1997.

[74] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[75] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[76] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[77] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *CVPR*, 2015.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[79] Y. LeCun, C. Cortes, and C. J.C. Burges. The MNIST database of handwritten digits. Technical report, 1998.

[80] J. Z. Leibo, J. Mutch, and T. Poggio. Why the brain separates face recognition from object recognition. In *NIPS*, 2011.

[81] A. Li, S. Shan, X. Chen, and W. Gao. Maximizing intra-individual correlations for face recognition across pose differences. In *CVPR*, 2009.

[82] S. Li, X. Liu, X. Chai, H. Zhang, S. Lao, and S. Shan. Morphable displacement field based image matching for face recognition across pose. In *ECCV*, 2012.

[83] Y. Li, B. Zhang, S. Shan, X. Chen, and W. Gao. Bagging based efficient kernel fisher discriminant analysis for face recognition. In *ICPR*, 2006.

[84] S. Liao, A. K. Jain, and S. Z. Li. Partial face recognition: Alignment-free approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[85] G. Lin, A. Milan, C. Shen, and I. D. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017.

[86] E. Littwin and L. Wolf. The multiverse loss for robust transfer learning. In *CVPR*, 2016.

[87] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. SphereFace: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[88] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[89] Y. Liu, F. Wei, J. Shao, L. Sheng, J. Yan, and X. Wang. Exploring disentangled feature

representation beyond face identification. In *CVPR*, 2018.

[90] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[91] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[92] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[93] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[94] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[95] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *CVPR*, 2016.

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

[97] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[98] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[99] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[100] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009.

[101] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[102] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *ICCV*, 2017.

[103] T. K. Perrachione, S. N. Del Tufo, and J. D. Gabrieli. Human voice recognition depends on language ability. *Science*, 333(6042):595–595, 2011.

[104] C. Qi and F. Su. Contrastive-center loss for deep neural networks. In *ICIP*, 2017.

[105] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[106] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[107] R. Ranjan, V. M. Patel, and R. Chellappa. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016.

[108] S. A. Rizvi, P. J. Phillips, and H. Moon. The FERET verification testing protocol for face recognition algorithms. In *FG*, 1998.

[109] J. Roth, X. Liu, and D. Metaxas. On continuous user authentication via typing behavior. *IEEE Transactions on Image Processing (TIP)*, 23(10):4611–4624, 2014.

[110] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *CVPR*, 2015.

[111] J. Rupnik and J. Shawe-Taylor. Multi-view canonical correlation analysis. In *Conference on Data Mining and Data Warehouses (SiKDD)*, 2010.

[112] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[113] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 Faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV workshop*, 2013.

[114] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv preprint arXiv:1604.05417*, 2016.

[115] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[116] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016.

[117] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen. Real-time rotation-invariant face detection with progressive calibration networks. In *CVPR*, 2018.

[118] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *BMVC*, 2013.

[119] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, and G. Lavrentyeva. Doppelganger mining for face representation learning. In *ICCV workshop*, 2017.

[120] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[121] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[122] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016.

[123] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[124] Y. Sun, D. Liang, X. Wang, and X. Tang. DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[125] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[126] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[127] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[128] O. Tadmor, Y. Wexler, T. Rosenwein, S. Shalev-Shwartz, and A. Shashua. Learning a metric embedding for face recognition using the multibatch method. In *NIPS*, 2016.

[129] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[130] W. Tan, B. Yan, and B. Bare. Feature super-resolution: Make machine see more clearly. In *CVPR*, 2018.

[131] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. In *CVPR*, 2015.

[132] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017.

[133] D. Y. Tsao, W. A. Freiwald, R. B. Tootell, and M. S. Livingstone. A cortical region consisting entirely of face-selective cells. *Science*, 311(5761):670–674, 2006.

[134] C. Turati, H. Bulf, and F. Simion. Newborns' face recognition over changes in viewpoint. *Cognition*, 2008.

[135] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *CVPR*, 1991.

[136] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016.

[137] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[138] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. NormFace: l_2 hypersphere embedding for face verification. *arXiv preprint arXiv:1704.06369*, 2017.

[139] H. Wang, Y. Wang, Z. Zhou, X. Ji, and W. Liu. CosFace: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[140] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang. Studying very low resolution recognition using deep networks. In *CVPR*, 2016.

[141] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[142] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 1987.

[143] X. Wu, R. He, Z. Sun, and T. Tan. A light CNN for deep face representation with noisy labels. *arXiv preprint arXiv:1511.02683*, 2015.

[144] Y. Wu, H. Liu, and Y. Fu. Low-shot face recognition with hybrid classifiers. In *ICCV workshop*, 2017.

[145] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim. Conditional convolutional neural network for modality-aware face recognition. In *ICCV*, 2015.

[146] X. Xiong and F. D. la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[147] D. Xu, S. Yan, D. Tao, S. Lin, and H.-J. Zhang. Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval. *IEEE Transactions on Image Processing (TIP)*, 16(11):2811–2821, 2007.

[148] Y. Xu, Y. Cheng, J. Zhao, Z. Wang, L. Xiong, K. Jayashree, H. Tamura, T. Kagaya, S. Pranata, S. Shen, et al. High performance large scale face recognition with multi-cognition softmax and feature retrieval. In *ICCV workshop*, 2017.

[149] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.

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

[151] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim. Rotating your face using multi-task

deep neural network. In *CVPR*, 2015.

[152] B. Yin, L. Tran, H. Li, X. Shen, and X. Liu. Towards interpretable face recognition. In *arXiv preprint arXiv:1805.00611*, 2018.

[153] X. Yin and X. Liu. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing (TIP)*, 2018.

[154] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017.

[155] X. Yu, B. Fernando, R. Hartley, and F. Porikli. Super-resolving very low-resolution face images with supplementary attributes. In *ECCV*, 2018.

[156] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *WACV*, 2014.

[157] H. Zhang, Y. Zhang, and T. S. Huang. Pose-robust face recognition via sparse representation. *Pattern Recognition*, 2013.

[158] L. Zhang, L. Lin, X. Liang, and K. He. Is faster R-CNN doing well for pedestrian detection? In *ECCV*, 2016.

[159] Q. Zhang, Y. N. Wu, and S.-C. Zhu. Interpretable convolutional neural networks. 2018.

[160] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tailed training data. In *CVPR*, 2017.

[161] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu. Random faces guided sparse many-to-one encoder for pose-invariant face recognition. In *ICCV*, 2013.

[162] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.

[163] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[164] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, et al. Towards pose invariant face recognition in the wild. In *CVPR*, 2018.

[165] Y. Zheng, D. K. Pal, and M. Savvides. Ring loss: Convex feature normalization for face recognition. In *CVPR*, 2018.

[166] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *CVPR*, 2016.

[167] 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 *CVPR*, 2015.

[168] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013.

[169] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: A deep model for learning face identity and view representations. In *NIPS*, 2014.

[170] W. W. ç. The model method in facial recognition. *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, 15(47):2, 1966.