# TOWARDS A ROBUST UNCONSTRAINED FACE RECOGNITION PIPELINE WITH DEEP NEURAL NETWORKS

By

Yichun Shi

A DISSERTATION

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

Computer Science – Doctor of Philosophy

2021

# ABSTRACT

## TOWARDS A ROBUST UNCONSTRAINED FACE RECOGNITION PIPELINE WITH DEEP NEURAL NETWORKS

By

Yichun Shi

Face recognition is a classic problem in the field of computer vision and pattern recognition due to its wide applications in real-world problems such as access control, identity verification, physical security, surveillance, etc. Recent progress in deep learning techniques and the access to large-scale face databases has lead to a significant improvement of face recognition accuracy under constrained and semi-constrained scenarios. Deep neural networks are shown to surpass human performance on Labeled Face in the Wild (LFW), which consists of celebrity photos captured in the wild. However, in many applications, e.g. surveillance videos, where we cannot assume that the presented face is under controlled variations, the performance of current DNN-based methods drop significantly. The main challenges in such an unconstrained face recognition problem include, but are not limited to: lack of labeled data, robust face normalization, discriminative representation learning and the ambiguity of facial features caused by information loss.

In this thesis, we propose a set of methods that attempt to address the above challenges in unconstrained face recognition systems. Starting from a classic deep face recognition pipeline, we review how each step in this pipeline could fail on low-quality uncontrolled input faces, what kind of solutions have been studied before, and then introduce our proposed methods. The various methods proposed in this thesis are independent but compatible with each other. Experiment on several challenging benchmarks, e.g. IJB-C and IJB-S show that the proposed methods are able to improve the robustness and reliability of deep unconstrained face recognition systems. Our solution achieves state-of-the-art performance, i.e. 95.0% TAR@FAR=0.001% on IJB-C dataset and 61.98% Rank1 retrieval rate on the surveillance-to-booking protocol of IJB-S dataset.

Dedicated to my parents

# ACKNOWLEDGMENTS

My deep gratitude first goes to my advisor, Dr. Anil K. Jain. Six years ago, I had the fortune to be admitted by Dr. Jain to his PRIP lab as a senior bachelor student. Since then, he has been giving valuable suggestions and guidance to both my research and life. In this lab, I have learned how to conduct research from looking for topics, conducting experiments to writing papers. I have also gained inspiration and skills from Dr. Jain that could be beneficial for my life. I appreciate his encouragement when I had troubles and the corrections when I made mistakes. His life-long enthusiasm for research has been, and will always be, a great inspiration for my career.

I thank all of the members of the PRIP lab for participating in my research and providing valuable feedback on my work. I will remember the joy that we shared and the discussions that we had.

I thank all the members in my PhD committee, namely Dr. Xiaoming Liu, Dr. Vishnu Naresh Boddeti, and Dr. Mi Zhang, for their valuable suggestions to my thesis work.

I thank my entire family for their love and support, especially my parents. Without them, I would not have the chance to come to MSU to become a PhD student.

I thank all my friends in East Lansing, who have been supporting me and keeping me company.

My thanks also go to all the other excellent scholars and fellow students whom I had chance to talk with or learned from in the past five years.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# KEY TO ABBREVIATIONS

**Acronyms / Abbreviation**

| | |
|---|---|
| FR | Face Recognition |
| SOTA | State-of-the-art |
| ROC | Receiving Operating Characteristic |
| TAR | True Acceptance Rate |
| FAR | False Acceptance Rate |
| CMC | Cumulative Match Characteristic |
| DIR | Detection & Identification Rate |
| PCA | Principal Component Analysis |
| LDA | Linear Discriminant Analysis |
| DNN | Deep Neural Network |
| CNN | Convolutional Neural Network |
| PFE | Probabilistic Face Embedding |
| GAN | Generative Adversarial Network |

# Chapter 1

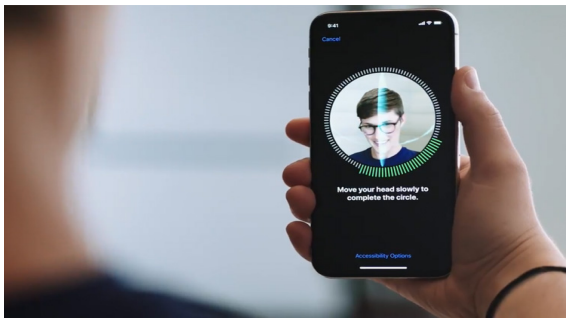
# Introduction

Face Recognition is a classic yet popular ongoing problem in the field of computer vision and pattern recognition. The general goal of Automatic Face Recognition (AFR) is to let the machine identify a person from his/her photos. Such a process could involve a set of typical challenges in computer vision problems: occlusion, illumination, out-of-plane rotation (pose change) and low image quality. On the other hand, AFR technology has a wide range of applications in forensics, access control, mobile payment, surveillance, etc, making it one of the most active research topics in the field of pattern recognition. In this chapter, we first review the applications of AFR systems and their development history, and then explain the pipeline of modern AFR systems and the challenges they face. Based on these challenges, we introduce our proposed methods and contributions.

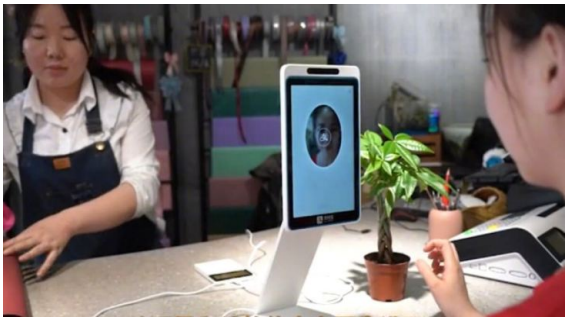
## 1.1 Applications of Automatic Face Recognition

The applications of face recognition can be classified into two types: face verification and face identification. Face verification systems, also known as 1:1 comparison, needs to decide whether two given face images (or collections) belong to the same person while a face identification (1:N comparison) system needs to identify (search) a probe image from a gallery set of N images. A more detailed explanation of face verification and face identification can be found in Section 1.5. Here, we list a few representative applications of face verification and identification.

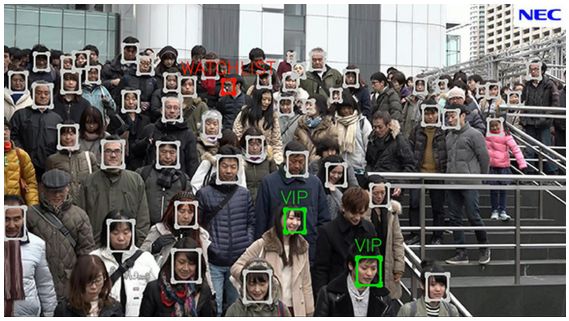
(a) Airport Security [9]




(b) iPhone X FaceID [10]



(c) Identification for Payment [11]




(d) Surveillance [12]

Figure 1.1 Example applications of face recognition.

### 1.1.1 Security

In many security-sensitive scenarios, we need to verify a person's identity for safety reasons. For example, as shown in Figure 1.1, many AFR systems have been deployed at airports world-wide to check the identity of a passport holder [13]. With more and more passengers taking international trips each year, such a system could significantly increase the efficiency of border control and reduce the burden on staff at airports. Similarly, many immigration checkpoints have also adopted face recognition systems to accelerate the passenger verification process.

### 1.1.2 Access Control

Another application of face verification is to check the access permission to certain buildings, devices or files stored in a computer. In corporate buildings, AFR systems are used to replace the traditional locks to control the entry gate. Starting with iPhone X, face recognition has been deployed as an alternative to PIN to secure the unlocking process (See Figure 1.1 (b)). Compared to

password and fingerprint, face recognition is more convenient to use since it only requires the user to look at the phone.

### 1.1.3 Identification

Besides face verification, another type of face recognition applications needs to identify a person from a large set of known people. For example, if the police have a photo of a criminal, they could use it to retrieve similar faces from a mugshot database to figure out potential identities of the criminal. Child trafficking is a severe problem in many developing countries. There, face identification can also be used to detect whether a child is reported to be lost [14] to solve such social problems. Besides security applications, face identification can also be applied to mobile payments (See Figure 1.1 (c)).

### 1.1.4 Surveillance

A special type of application of face identification is associated with surveillance cameras. These surveillance cameras play a key role in the management of mega-cities across the world. Till 2019, there were estimated to be 770 million surveillance cameras installed around the world [15]. However, effectively utilizing these surveillance videos is not a simple task, since a large amount of human labor would be needed to monitor or review them. In contrast, a robust face detection and identification algorithm could go through massive number of videos to localize potential criminals and operate 24 hours a day seven days a week.

## 1.2 The Development of Automatic Face Recognition

### 1.2.1 Traditional Solutions

Since the first study on AFR by Takeo Kanade [16] in 1970s, the technology of automatic face recognition has evolved drastically. Many different approaches have been explored to represent

and compute similarity two face images under consideration. In the early stages, methods that explicitly model the geometric shape or texture of faces were used to represent the face images. For example, Active Shape Models [17] represent a face by the coordinates of facial landmarks. However, such methods are limited in terms of representation power and are sensitive to the variations that could appear in face images, such as pose, illumination, and expression (PIE). Subspace-based representations, such as EigenFace [18] and FisherFace [19], have been proposed to model faces images by a set of basis components in a linear subspace. Each image is represented by the coefficient of the bases, which can be further used for recognition tasks. Later, manually designed visual descriptors, such as Scale-invariant Feature Transform (SIFT) [20] and Local Binary Patterns (LBP) [21], became popular in computer vision tasks. These features are shown to be effective on face recognition tasks as well [21] and achieves even better performance when combined with data-driven methods [22], such as Linear Discriminant Analysis (LDA) [19] and JointBayes [23].

### 1.2.2 Deep Face Recognition

In recent years, due to the advent of large-scale web data and efficient parallel computing devices, i.e. GPUs, Convolutional Neural Networks (CNNs), as a pure data-driven method, has replaced traditional methods and achieved impressive performance on a wide range of computer vision tasks, such as image classification [24], detection [25] and segmentation [26]. A series of CNN-based face recognition algorithms have been proposed since 2014, including DeepFace [27], DeepID series [28, 29] and FaceNet [30]. These algorithms not only outperform traditional methods by a large margin, but they beat human beings on face verification tasks [28, 31]. Compared to LBP and LDA, CNN are able to learn a much more complicated non-linear mapping function to serve as the feature extractor, which is more discriminative and less sensitive to different facial variations. After the early work on CNN-based face recognition, subsequent studies have explored different loss functions to improve the discrimination power of the feature representation. Wen et al. [32] proposed center loss to reduce intra-class variation. A series of works have also proposed to use metric learning for face recognition [30, 33]. Recent efforts have attempted to achieve discriminative

embeddings with a single identification loss function where proxy or prototype vectors are used to represent each class in the embedding space [34, 35, 36, 37, 38].

### 1.2.3 From Constrained to Unconstrained Face Recognition

Along with the boost in AFR algorithms, the evaluation datasets and protocols for AFR systems have been updated frequently. In the early studies, the input face photos to be recognized are mostly captured under constrained settings, where there is a limited amount of variation in terms of pose, illumination and expression (PIE). For example, the Yale-B face dataset [39] released in 2000 only consists of gray-scale frontal faces with different illumination. In 2007, the Labeled Faces in the Wild (LFW) [3] dataset was released. As the first benchmark composed of face images captured in the wild (not under controlled settings), LFW became a major challenge to the AFR systems before the deep learning era (see Figure 1.3). While traditional approaches [22] achieved 95.17% accuracy on LFW under the standard verification protocol, CNNs that were trained on large-scale datasets quickly saturated the benchmark with performance higher than 99.00% [31, 30]. Since then, more unconstrained benchmarks have been released to evaluate the performance of FR algorithms. For example, NIST released three benchmarks developed under IARPA Janus program [40], namely IJB-A [5], IJB-B [41], and IJB-C [42], that are composed of a mixed set of celebrity photos along with video frames. Since the faces in these images are manually cropped by humans rather than off-the-shelf face detectors, the faces in these datasets could include arbitrary PIE variations. Further, instead of a closed-set recognition, the models today are required to perform recognition in an open-set setting, i.e. the test (query) subjects may not be present in the database of known subjects (gallery), which makes the tasks even more difficult. In spite of these challenges, deep neural networks quickly saturated even these benchmarks by learning from larger and larger training datasets and newer models. Recently, researchers have been focusing on more challenging cases, such as surveillance face recognition [1] and low-resolution faces in the wild [6], which represent a more realistic setting in real-world applications. More details on these datasets can be found in Section 1.5.

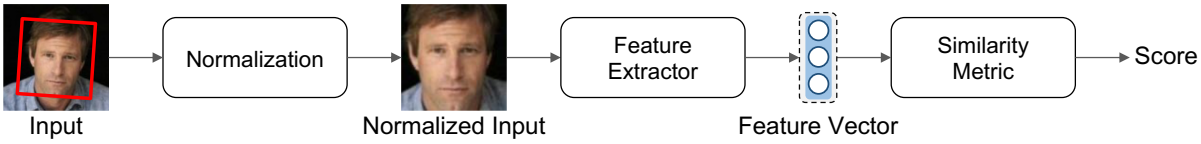



Figure 1.2 The pipleline of automatic face recognition systems. Here, we assume the face images are already detected and hence omit the detection step.

## 1.3 Pipeline of Automatic Face Recognition

As shown in Figure 1.2, the pipeline of AFR typically includes three steps: normalization, feature extraction and comparison. Here, we assume the faces have already been detected and we do not discuss it.

**Normalization** In this step, spatial transformations are conducted to reduce the facial variations before sending the input image to the feature extractor module. Different methods can be used to reduce such variations. The simplest solution is to use the location of bounding box or landmarks to crop a canonical view of the input face [30, 43]. Some use more complicated 3D models to frontalize the face to further reduce the variation [27]. The most common solution is to detect 5 landmarks (eyes, nose and mouth corners) and apply a similarity transformation [32, 34].

**Feature Extraction** In this step, either a manually designed or a learned representation are used to extract the discriminative features from the faces. Both the LBP descriptor and LDA methods mentioned in last section belong to this step. Today, almost all ongoing research use a CNN as the feature extractor, which maps an RGB image to a feature vector with fixed length. The CNN is first trained on a large-scale web-crawled database, with millions of face images covering hundreds of thousands of identities [44, 2], and then the output vectors of its hidden layers are used as the extracted features to represent the faces. The network is trained either with metric learning loss functions [30] or classification tasks [27].

**Similarity Metric** The choice of similarity metric mainly depends on the representation. For example, for histogram-based features, such as LBP, $\chi^2$ measure is used to compute the distance. Assuming the features are generated by a Gaussian distribution, Chen et al. [23] proposed to use a joint formulation of concatenated feature vectors to compute the facial similarity, which is also shown to be effective on deep representations [45]. The most widely adopted similarity metric for deep face representations is cosine similarity. This is mainly because the hidden features learned by the neural networks with a classification loss are distributed in a radial way [46, 34].

## 1.4 Challenges in Unconstrained Face Recognition

The major challenges of unconstrained face recognition, compared with constrained ones, lies in the large facial variations, including pose, illumination, expression, low resolution, occlusion, etc. Although illumination and expression were considered challenging for manually designed features, deep representations that are trained on large-scale web datasets turn out to be relatively invariant to such variations [29]. Thus, they are no longer a focus in recent studies. Different methods have been proposed to learn pose-invariant deep face representations. Some use pose labels during training to learn a pose-disentangled deep feature [47, 48] while others have utilized 3D models to build deep models that could frontalize the face images before feature extraction [49]. On the other side, state-of-the-art (SOTA) face recognition models [38] that are trained on generic face datasets with deeper models and margin-based loss functions have also been shown to perform well on cross-pose face verification tasks [4]. Compared with other types of variations, low resolution and occlusion are more difficult because they imply the loss of information in the input faces. While deep representations have achieved human-level performance on constrained face photos, many evaluation benchmarks have been released to evaluate deep face recognition models on surveillance and web videos [1, 6], where lower resolution and occlusion are the main challenges.

In a complete AFR system, the individual impact of aforementioned facial variations depends on the choice of different modules in the FR pipeline. Here we briefly introduce the connection

between different facial variations and the modules in an AFR pipeline, which further motivates our proposed methods in Chapters 2-5:

- The face normalization step is mainly correlated with the cross-pose face recognition performance. A more complicated normalization method, e.g. 3D modeling, could significantly alleviates the pose variation problem. However, under unconstrained settings, given a low quality image, the 3D model might not be able to accurately reconstruct the structure of the input face. In Chapter 2, we introduce an attention-based learning framework that could exploit local features from faces of different poses to handle this issue.

- In Chapter 3, we show that the common choice of similarity metric, i.e. cosine similarity between embedded vectors, could suffer from facial variations that cause information loss, such as low resolution and occlusion. We propose an uncertainty-based representation to solve this problem.

- The representation learning step is related to all kinds of variations. The performance differs depending on what kind of dataset and loss function we choose to train the model with. Thus, a trade-off often exists when the performance degrades on a certain type of data when we fit our model to handle other types of variations. In Chapter 4, we propose a universal representation learning framework that is able to simultaneously improve feature discrimination power with different variations. In Chapter 5, we further propose a semi-supervised representation learning framework that utilizes an auxiliary unlabeled dataset to augment the labeled training data to improve the generalizability of the face embeddings.

## 1.5 Evaluation Metrics and Datasets

### 1.5.1 Evaluation

The tasks of face recognition can be concluded as two types: face verification and face identification or search.

In **face verification**, also known as 1:1 comparison, the system is required to determine whether a pair of face images belong to the same subject by applying a threshold to the similarity score. Two types of metrics are used here for evaluating face verification protocol. The first metric is the accuracy:

$$Accuracy(N, T) = \frac{\text{Number of correct comparison at threshold, } T}{\text{Number of all possible pairs, } N}. \tag{1.1}$$

This metric is usually used for protocols with balanced number of positive and negative pairs, such as in LFW [3] and CFP [4]. The threshold $T$ is usually determined under a cross-validation protocol. The second metric, which is closer to a real-world verification requirement is to evaluate the True Accept Rate of input pairs at a fixed False Accept Rate. Formally, this metric is defined as:

$$TAR(N_p, T) = \frac{\text{Number of accepted genuine pairs at threshold, } T}{\text{Number of all genuine pairs, } N_p}. \tag{1.2}$$

$$FAR(N_n, T) = \frac{\text{Number of accepted impostor pairs at threshold, } T}{\text{Number of all impostor pairs, } N_n}. \tag{1.3}$$

To achieve a lower FAR, one would like to lower the threshold $T$, which would cause a lower TAR, too. Therefore, we can evaluate the performance by choosing the threshold based on a desired FAR value.

In **face identification**, also known as 1:N comparison, the system is given a gallery set with images of known identities. Then, given a probe face image, the system needs to determine which person in the gallery the input face belongs to. In particular, depending on whether there exists non-mate probes (whose corresponding subject is not in the gallery), the identification protocol can be further categorized into closed-set identification and open-set identification. In the open-set identification, the system needs to first determine whether the input face identity is in the gallery before trying to identify him/her. For closed-set identification, rank retrieval rate is used to evaluate the performance as shown below:

$$RetrievalRate(N, K) = \frac{\text{Number of successfully retrieved probes with top } K \text{ returns}}{\text{Number of all probes } N} \tag{1.4}$$

and True Positive Identification Rate (TPIR) at False Positive Identification Rate is used to report the performance of open-set face recognition benchmarks:

$$TPIR(N,T) = \frac{\text{Number of retrieved mate probes with score above threshold } T}{\text{Number of all mate probes } N} \tag{1.5}$$

$$FPIR(N,T) = \frac{\text{Number of non-mate probes with score above threshold } T}{\text{Number of all non-mate probes } N} \tag{1.6}$$

Similar to TAR@FAR for verification, the threshold $T$ here is chosen depending on the FPIR. For probes with a mate in the gallery, a probe's mate is considered to be successfully retrieved only if it is returned as top-1 result.

### 1.5.2 Datasets

Here, we briefly introduce all the datasets that we use for training and evaluating the AFR systems. Since all the models in our work are based on deep neural networks, which require a large number of images to fit the parameters, we use two public web-crawled datasets for training:

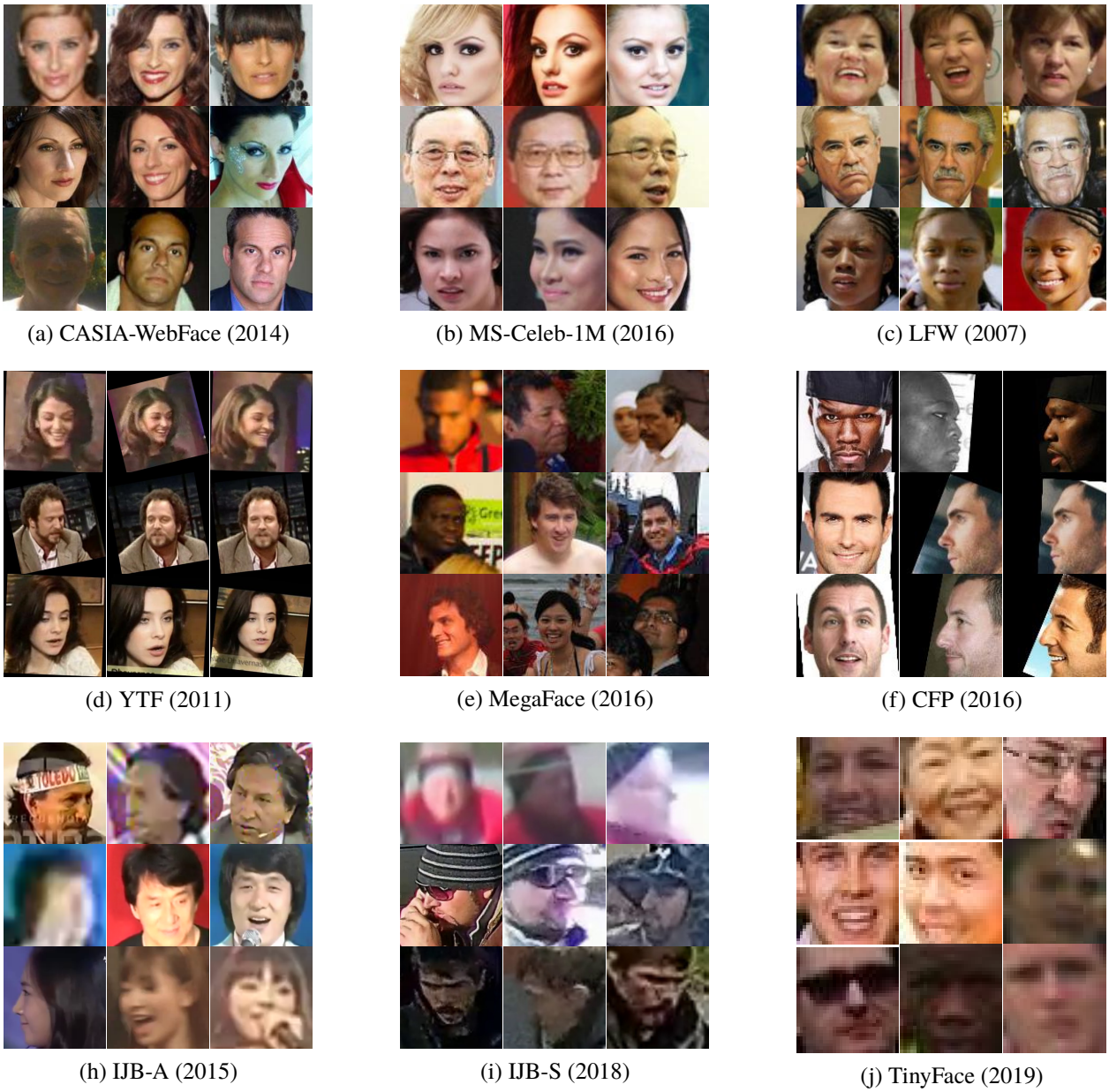
**CASIA-Webface** [50] contains about 0.5M high-quality celebrity photos of 10,575 subjects captured "in the wild". All the face images are collected from internet by searching celebrity names.

**MS-Celeb-1M** [2] contains 8M face photos of about 85K subjects. The images are collected in a similar way as CASIA-Webface. However, the original MS-Celeb-1M contains a large number of mislabeled images. Therefore, a cleaned version is usually used instead of the original one. In this thesis, we use a publicly available clean list[1] and the clean list from ArcFace [38] for the experiments.

We show some example images from MS-Celeb-1M in Figure 1.3 (a). The images in CASIA-Webface are similar to them. For evaluation, we consider 8 different benchmarks, whose images present different types and degrees of facial variations:

**LFW** [3] contains 13,233 near-frontal and high-quality face photos of 5,749 subjects. The verification protocol used in this thesis includes 6,000 face pairs.

[1] `https://github.com/inlmouse/MS-Celeb-1M_WashList`.



(a) CASIA-WebFace (2014) (b) MS-Celeb-1M (2016) (c) LFW (2007)

(d) YTF (2011) (e) MegaFace (2016) (f) CFP (2016)

(h) IJB-A (2015) (i) IJB-S (2018) (j) TinyFace (2019)

Figure 1.3 Example images of six representative datasets. The images are sampled from MS-Celeb-1M [2], LFW [3], CFP [4], IJB-A [5], IJB-S [1] and TinyFace [6] respectively.

**YTF** [51] contains 3,425 videos of 1,595 subjects. The verification protocol used in this thesis includes 5,000 video pairs.

**MegaFace** [52] contains 1M face images from Flicker as distractors. The FaceScrub dataset is used as the probe set in our experiments, which contains 3,530 high-quality face images of 80 subjects.

**CFP** [4] contains 7,000 frontal/profile face photos of 500 subjects. We only test on the frontal-profile

(FP) protocol, which includes 7,000 pairs of frontal-profile faces.

**IJB-A** [5] is a template-based benchmark, containing 25,813 faces images of 500 subjects. Each template includes a set of still photos or video frames. Compared with previous benchmarks, the faces in IJB-A have larger variations and present a more unconstrained scenario.

**IJB-C** [42] is an extension of IJB-A with 140,740 faces images of 3,531 subjects. The verification protocol of IJB-C includes more impostor pairs so that we can compute True Accept Rates (TAR) at lower False Accept Rates (FAR).

**IJB-S** [1] is a surveillance video benchmark containing 350 surveillance videos spanning 30 hours in total, 5,656 enrollment images, and 202 enrollment videos of 202 subjects. Many faces in this dataset are of extreme pose or low-quality, making it one of the most challenging face recognition benchmarks.

**TinyFace** [6] is a dataset to evaluate the face recognition models on low-resolution face images. The dataset contains 5,139 labelled facial identities given by 169,403 natural low-resolution face images. Closed-set identification rate is used to evaluate the systems on this benchmark.

Example images from some of these datasets are shown in Figure 1.3.

## 1.6 Dissertation Contributions

The main contributions of this dissertation are as follows:

- A spatial transformer-based attention module that automatically detects salient facial regions to extract local features. The attention module could be trained without labels.
- A framework that efficiently combines multiple region attention modules to extract local features and incorporates them into global facial representation. Experimental results on unconstrained face databases show that the method could effectively boost the performance of a base face matcher when more salient regions are combined.
- A new type of face representation that takes feature uncertainty into account. Given a

pre-trained deterministic deep face embedding, the proposed method could convert it into a probabilistic face embedding (PFE) by representing each face image as a distribution in the latent space. The probabilistic embedding adds additional interpretability to deep face representations and can be used as a quality assessment method to control the enrollment of face images.

- A probabilistic method that effectively utilizes data uncertainty to combine and compare different probabilistic face embeddings.

- An universal feature learning framework that learns a set of sub-embeddings to tackle different variations in unconstrained face recognition. A confidence-controlled face identification loss and variation-based decouple loss are proposed to regularize the facial features to handle multiple variations. Experiments show that the proposed method could incrementally enhance the feature representations when more types of variations are introduced into the training data. Combining decoupled sub-embeddings with PFE leads to SOTA performance on several challenging face recognition benchmarks.

- A semi-supervised feature learning framework that incorporates an auxiliary unlabeled dataset into the training of deep face embeddings. A generator is trained to automatically discover the latent styles in the unlabeled dataset such that it can be used to augment the labeled dataset. Then, we can jointly regularize the embedding model from both the image space and the feature space to improve its generalizability.

## 1.7 Thesis Structure

Ch. 2 of this thesis presents a framework of enhancing global face features with local information. spatial transformers are used as attention modules to automatically localize salient facial regions to extract local features, which are then fused into the holistic features. Ch. 3 presents a new type of face representation, namely Probabilistic Face Embeddings (PFEs). PFEs incorporate data

uncertainty into face representations and are able to improve face recognition performance by taking uncertainty into account during template fusion and template comparison. In Ch. 4, we propose a framework to learn a universal face representation. Different types of data augmentation are combined to mimic a setting where one has access to a large training dataset of unconstrained faces and new loss functions are proposed to learn decoupled features from difficult training samples. Ch. 5 further studies the possibility of using an unlabeled dataset to augment a labeled training set in terms of diversity, where we show improved model generalizability to unconstrained faces. The last chapter discusses the conclusions of this dissertation and presents directions for future work. The experimental results of the work in this thesis were previously presented in [53, 54, 55, 56].

# Chapter 2

# Learning Local Face Features with Visual Attention

## 2.1 Introduction

As shown in Figure 1.2, most AFR systems adopt a normalization step for pre-processing to ensure the input faces are in a similar position and orientation, reducing the intra-class variations and making the recognition task easier [27, 31, 50, 30, 57]. However, as the complexity of unconstrained face images increase, even though aligned, 2D face images can still appear very differently, as shown in Figure 2.1. As such, constructing global face models becomes a very difficult task. Because of this difficulty, an attractive idea is to model different facial parts individually and combine them to generate a global representation. Recognizing complex objects by their parts is a popular technique in pattern recognition. In the well-known Deformable Part Model (DPM) [58], different part filters are learned and combined with a root filter to detect complex objects in the images efficiently. Similar ideas, such as decomposing faces into different parts, have been shown to work well for face detection [59, 60, 61]. A highly successful, parts-based face recognition approach, called the DeepID series [28, 29, 31], cropped a large number of different local patches either at fixed positions or around landmarks in the face image, trained a single deep convolutional network on each of these

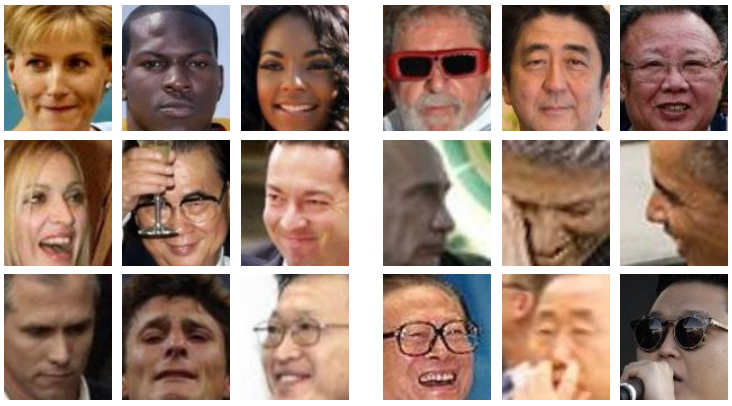



Figure 2.1 Example images in LFW and IJB-B after alignment using MTCNN [7]. The image in the first row are well aligned and all the facial parts are located in a consistent way. The face images in the second and third rows, although aligned, still appear in a quite different way because of large pose variations or occlusion.

regions, and fused the representations from all the networks by training on a validation dataset. The success of works like DeepID indicate that although face is a nearly rigid object, building models for different face regions can also help improve the performance of face recognition systems.

One of the most important problems in parts-based face recognition approaches, is the localization of the target parts. In other words, although the faces are aligned, parts of a face shown in a fixed region could be quite different for different people at different poses, which reduces the discrimination ability of these parts-based models. One approach to solving this problem is to use the detected landmarks to crop rectangular patches around those respective landmarks. However, even with these landmarks, it is still difficult to decide what regions we should crop since some regions may be useful for recognition, and others may not. Given this difficulty, we turn to another technique to find and localize discriminative regions automatically that has become popular in the vision community, i.e. visual attention mechanism [62, 63, 64, 65].

By using a differentiable visual attention network, we can build an end-to-end system where the global recognition network and several parts-based networks are trained simultaneously. In this proposed end-to-end system, a fully connected layer for fusing features can be trained together with the recognition networks, which helps the sub-networks to explore more discriminative

features complementary to the global representation. In addition, the visual attention network learns to localize distinct local regions automatically without any landmark supervision. Our experiments show that the proposed approach can further improve the state-of-the-art networks on challenging benchmarks such as IJB-A and IJB-B. More concisely, contributions of this chapter can be summarized as follows:

- We designed an end-to-end face recognition system including global network, parts-based networks, attention network and a fusion layer that are trained simultaneously.
- We showed that discriminative regions can be be localized automatically without using facial landmarks by using a visual attention network.
- We showed that adding parts-based networks can further improve the performance of state-of-art deep networks on challenging protocols, including BLUFR, IJB-A and IJB-B, with little complexity increase.

## 2.2 Related Work

### 2.2.1 Parts-based Deep Face Recognition

Our proposed approach is predominantly inspired by the success of the DeepID series [28, 29, 31]. In their first work [28], ten different regions were cropped, respectively, from a face image (five large regions at fixed positions and five small regions around detected landmarks). For each region, RGB and gray-scale patches of five different scales were generated and each trained with a single convolutional neural network to output a feature vector of 160 dimensions. The features were then concatenated and the dimensionality was reduced with additional training on a validation set. In DeepID2, 400 patches at different positions, scales, color channels and horizontal flipping were cropped and used for training 200 different networks. After feature selection, 25 patches were selected to extract a $4,000$-dimensional feature vector, which was finally reduced to 180-dimensional vector

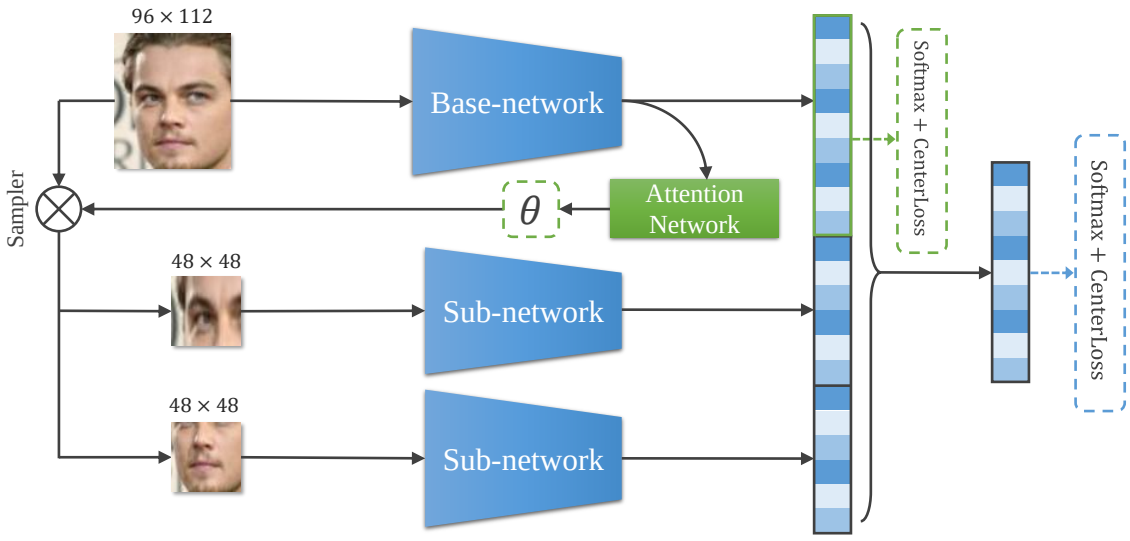



Figure 2.2 An example architecture of the proposed end-to-end network with $K = 2$ sub-networks. A $96 \times 112$ image is first fed into the base-network, which is a single CNN for face recognition. The feature map of the last convolutional layer of the base-network is then both used to learn a global representation with a fully connected layer, and $K$ transformation matrices with an attention network of two-stacked fully-connected layers. The regions of interest are sampled into patches of size of $48 \times 48$. $K$ smaller CNNs as sub-networks follow to learn local features from these automatically localized patches. All the global and local features are then concatenated and fused by another fully connected layer.

with PCA. The authors showed that combining these features from different regions substantially improved the face recognition performance.

### 2.2.2 Visual Attention Network

Visual attention is a mechanism to automatically localize objects of interest in an image or parts of an object. Ba et al. [62] used a recurrent attention model to locate the objects in order to better perform multi-object classification. A similar scheme was used in [63] to generate captions for images. Xiao et al. [66] proposed to use visual attention proposals for fine-grained object classification by clustering the channels of a feature map into different groups and generating patches based on the activation of individual groups. In [64], a recurrent structure of a CNN and attention proposal network is proposed to zoom into small regions for fine-grained classification. The input of the attention network is the feature map of the last convolutional layer rather than raw images so that the

computational cost can be reduced. We adopt a similar strategy in our network. Only two levels of CNNs are used in our approach but more than one patch is generated by the attention network. In addition, we use Spatial Transformers [65], which use a projective transformation matrix $\theta$ to transform the original input image, enabling us to better sample patches. By multiplying $\theta$ and the coordinates of pixels in the output image, the spatial transformer computes the corresponding coordinates of each pixel in the input image, and samples them through bi-linear interpolation. This transformer is differentiable, allowing the attention network to be learned end-to-end without labels. In [65], experiments showed that the spatial transformer network is able to automatically localize distorted digits, and street view house numbers. Subsequently, the performance of fine-grained classification is improved by generating multiple region proposals. Finally, Zhong et al. [67] showed that by training an attention network with spatial transformers, an end-to-end face recognition network which automatically learns the alignment can achieve comparable results to those with pre-aligned images.

## 2.3 Approach

In this section, we outline an end-to-end network which includes a *base-network* for learning a global representation from the whole face image, several *sub-network*s for modeling specific facial parts, an attention network for generating region proposals to feed into the sub-networks and a fusion layer to fuse the global and local features.

### 2.3.1 Overall Architecture

A graphic illustration of the overall architecture is shown in Figure 2.2. The input image size is $96 \times 112$. The proposed network begins with a base-network which can be any single convolutional neural network for face recognition. In particular, we employ the Face-ResNet proposed in [68] because of its good generalization ability and its state-of-the-art performance. In order to reduce the computational cost of the attention network, we adopt a similar approach as [64], where the attention

Table 2.1 The architecture of the attention network.

| Type | Output Size |
|---|---|
| Batch Norm + Fully Connected | 128 |
| Batch Norm + Fully Connected | $8 \times K$ |

network is connected to the last hidden convolutional layer rather than the input image. The attention network outputs $K$ projective transformation matrices $\theta$, each of which has 8 parameters. Here, $K$ is a hyperparameter. For each of the $K$ transformation matrices, a spatial transformer is used to sample a $48 \times 48$ patch from the region of interest via bi-linear interpolation. The sampled patch is then used by a smaller sub-network to learn local features. The global representation is of 512 dimensions, while the length of each local feature vector is 128 dimensions. All of them are concatenated together and fused by a fully connected layer to generate a 512-dimensional representation.

A softmax layer is added to both the global representation and the fused representation for classification in the training phase. Notice that the gradient is not propagated back through the fusion layer to the global representation. This allows the base-network to be trained independently, and it encourages the sub-networks to explore new features complementary to the global representation. Experimental result shows that such an approach enables the model to converge faster and leads to better generalizability. The softmax mainly learns to scatter the features of different classes, which is correspondent to the inter-class dissimilarity. Therefore, in order to reduce the intra-class variation, we also adopt the center loss proposed in [32] with the recommended setting of $\alpha = 0.5$ and $\lambda = 0.003$. The center loss is applied to both the global representation and fused representation.

### 2.3.2 Attention Network

Details about the attention network are shown in Table 2.1. Because the input to this network is the feature map of the last convolutional layer of the base-network that contains rich semantic information, the attention network is composed of only two fully-connected layers, saving a large amount of computational resources. We add a batch normalization layer [69] along with a ReLU

Table 2.2 The architecture of the sub-networks.

| Type | Output Size | Filter Size/Stride |
|---|---|---|
| Convolution | $48 \times 48 \times 32$ | $3 \times 3/1$ |
| Convolution | $48 \times 48 \times 64$ | $3 \times 3/1$ |
| Max Pooling | $24 \times 24 \times 64$ | $2 \times 2/2$ |
| Convolution | $24 \times 24 \times 64$ | $3 \times 3/1$ |
| Convolution | $24 \times 24 \times 128$ | $3 \times 3/1$ |
| Max Pooling | $12 \times 12 \times 128$ | $2 \times 2/2$ |
| Convolution | $12 \times 12 \times 96$ | $3 \times 3/1$ |
| Convolution | $12 \times 12 \times 192$ | $3 \times 3/1$ |
| Max Pooling | $6 \times 6 \times 192$ | $2 \times 2/2$ |
| Convolution | $6 \times 6 \times 128$ | $3 \times 3/1$ |
| Convolution | $6 \times 6 \times 256$ | $3 \times 3/1$ |
| Fully Connected | 128 | |

activation layer [70] both before and after the first fully-connected layer to accelerate the training of attention network. The second fully connected layer outputs $K$ transformation matrices. Then a spatial transformer module is used to sample the corresponding partial regions according to each of these matrices. Finally, there are several implementation subtleties to note.

First, because we are using a projective transformation, the sampled region is not restricted to be a rectangular shape. This means that the original image could be warped. However, Zhong et al. [67] showed that a better performance can be achieved with a projective transformation than a similarity transformation for face alignment. One plausible explanation for this is that neural networks do not perceive images in the same way as human do. As such, networks are able to learn better features from warped images.

Second, we multiply the learning rate of the attention network by 0.0001. Without performing this scaling, the output transformation deviates too much before the network is able to learn a set of reasonable parameters.

Third, the weights of the last fully connected layer are initialized as zero, while its biases are initialized as the flatten vector of the initial K transformation matrices. In experiments, we use manual initialization for these matrices if $K$ is small and random initialization if $K$ is large.

### 2.3.3 Sub-Network for Modeling Facial Parts

Since the information in a local region is relatively small, it would be unnecessarily complex to use a network with as many parameters as the base-network to learn representations from these patches. As such, we use a simple architecture for all the sub-networks, as shown in Table 2.2. It is very similar to the network used in [50] except that it uses fewer layers. We add a fully connected layer at the end of the sub-network to learn a compressed local feature vector. Finally, we add a batch normalization along with a ReLU layer after every convolution and fully connected layer. Because the sub-networks take a smaller input and have fewer parameters compared with base-network, they only add little extra run-time to the whole model, as shown in 2.4.1.

### 2.3.4 Promoting Sub-networks for Feature Exploration

Although theoretically the larger the number of sub-networks, the more complementary local features can be learned to improve the robustness of the fused representation, we find that the improvement of the performance after adding a large number of sub-networks is usually negligent. An explanation for this is found by the magnitude of the weights in the fusion layer for each dimension in the concatenated feature. Figure 2.3 shows that many local features have very small weights in the fusion layer. This indicates that there are some sub-networks which contribute little to the final fused representation. Additionally, this could diminish the loss propagated back to the base-networks and prevents the sub-networks from learning efficiently. As such, some sub-networks become "dead" during training. Therefore, inspired by [71], we add a promotion loss to explicitly promote the weights in the fusion layer for those local features. Notice that in [71], the promoted parameters are those related to a certain output class, however, in our case they are those related to a certain input dimension. In particular, let's denote an input feature vector as $\mathbf{x} = [\mathbf{x}^g, \mathbf{x}^l]$ where $\mathbf{x}^g$ is the global feature vector and $\mathbf{x}^l$ is the vector of all local features concatenated into one column. The fused representation $y$ is obtained with a fully connected layer $y = W\mathbf{x} + \mathbf{b}$. Corresponding to $x^g$ and $x^l$,

$W$ can be viewed as the concatenation of two matrices $W^g$ and $W^l$, where

$$y = W^g \mathbf{x}^g + W^l \mathbf{x}^l + \mathbf{b} \tag{2.1}$$

The goal of the proposed promotion loss $L_p$ is to encourage the local weights to be similar to the global weights:

$$L_p = \frac{1}{D_l} \sum_{i=0}^{D_l} \left\| \left\| W_i^l \right\|^2 - \alpha \right\|^2, \tag{2.2}$$

where

$$\alpha = \frac{1}{D_g} \sum_{i=0}^{D_g} \left\| W_i^g \right\|^2 \tag{2.3}$$

and $D_l$, $D_g$ refer to the number of dimensions in the local and global feature vectors, respectively. $W_i^l$ refers to the $i_{\text{th}}$ column of $W^l$, similar for $W_i^g$. The promotion loss is added as a regularization loss with coefficient $\lambda$. As shown in 2.3, after adding promotion loss, the distribution of the weights in the fusion layer become much more uniform, thus avoiding the problem of “dead" sub-network and encouraging the sub-networks to find more discriminative features.

## 2.4 Experiments

### 2.4.1 Implementation Details

. We conduct all of our experiments using Tensorflow 1.2. First, we implement the Face-ResNet in [68]. We follow the same settings for the learning rate and center loss. All the images are first aligned using landmarks detected using MTCNN [7] and trained on the CASIA-Webface dataset [50]. The resulting network achieves a verification accuracy of 98.77% on the standard LFW protocol. This result is quite comparable to the performance originally reported in [68], however, we do note a slight drop in performance (from 99.00% to 98.77%). The most plausible explanation is that we are using a different library for implementation. All the following experiments are compared to this baseline result.

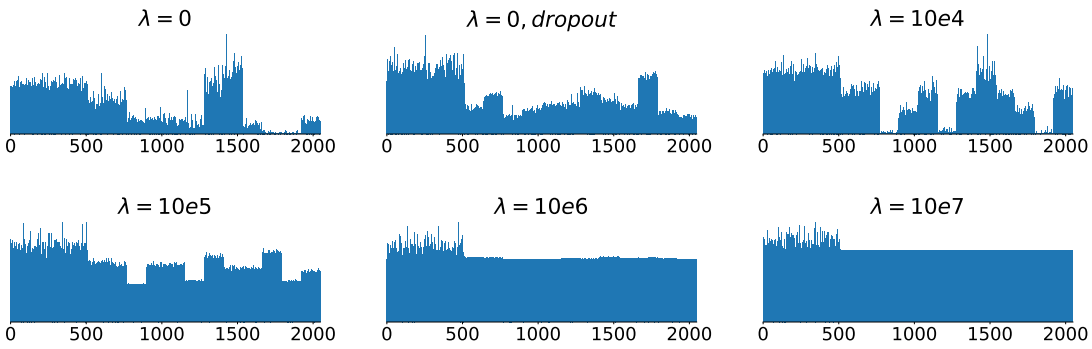



Figure 2.3 Magnitude of the weights of the fusion layer over different input dimensions when using different $\lambda$ for the promotion loss. Without promotion loss, many dimensions have little weight, resulting in “dead" sub-networks. Dropout helps to promote the weights, but diminishes the performance.

For the sub-networks, we adopt two schemes to initialize the transformation matrices $\theta$:

- **Model A**: a small network with $K = 3$ rectangular regions initialized in the upper, middle and bottom face, respectively.
- **Model B**: a relatively larger network with $K$ randomly initialized rectangular regions, whose widths and heights are between 30% and 60% of the original image.

The reason that we manually initialize Model A is that when $K$ is rather small, the randomly initialized regions are not guaranteed to be distributed well. For example, they may have a large amount of overlap and only cover a small part of the entire face image. This would result in leaving behind crucial information useful for recognition. Therefore, we manually choose three rectangular regions that cover different parts of the face for Model A.

We follow the same training settings as [68] with a batch size of 256 and 28,000 training steps. The promotion loss weight is set to $\lambda = 10^5$ based on the results of a grid search. We use two Nvidia Gefore GTX 1080 Ti GPUs to train Model A and four for Model B. As for time complexity, there is only a slight increase in run-time: for base-network, Model A and Model B, it takes 0.003s, 0.003s and 0.004s per image to extract features with one GPU, respectively.

In order to evaluate the proposed method and implementation, we first study the effectiveness of the proposed modules using LFW dataset with both standard and BLUFR protocol [8]. Then we

Figure 2.4 Example pairs that are misclassified by base-network but are classified correctly on LFW dataset. Pairs in the green box are genuine pairs and pairs in the red box are impostor pairs. We use the average threshold of BLUFR [8] face verification for VR@FAR= 0.1% on 10 splits.

evaluate the proposed model on more challenging IJB-A [5] and IJB-B [41] benchmarks. Because the purpose of this chapter is to present a system to improve any face recognition network instead of achieving the best result on these specific protocols, and since most results on the benchmarks are based on different architectures and training datasets, we believe it is not fair to compare the absolute performances. Thus, we only compare the relative performance of the proposed system with the original base-network.

### 2.4.2 Evaluation of Proposed Modules on LFW

In the proposed network, we use an attention network to localize $K$ discriminative regions rather than cropping a fixed patch, train a fusion layer to compress the concatenated feature and add promotion loss encouraging the sub-networks to explore more discriminative features. Here we evaluate the effectiveness of these modules by comparing the results with and without these modules on two protocols on LFW dataset: standard and BLUFR [8]. The standard verification protocol of

Table 2.3 Evaluation results of the proposed model with/without certain modules on standard LFW and BLUFR protocols. “AN” means "Attention Network"; “FL” means "Fusion Layer"; "PL" refers to "Promotion Loss". “Y” indicates the module is used while “N” indicates that module is not used. Accuracy is tested on the standard LFW verification protocol. Verification Rate (VR) and Detection and Identification Rate (DIR) are tested on the BLUFR protocol.

| Type | AN | FL | PL | Accuracy | VR @FAR= 0.1% | DIR Rank-1 @FAR= 1% |
|---|---|---|---|---|---|---|
| Base-net | | | | 98.77% | 94.96% | 72.96% |
| Model B | N | Y | Y | 98.67% | 95.54% | 74.33% |
| Model B | Y | N | Y | 98.78% | 95.63% | 76.37% |
| Model B | Y | Y | N | 98.75% | 95.83% | 75.75% |
| Model A | Y | Y | Y | 98.85% | 95.90% | 77.51% |
| Model B | Y | Y | Y | **98.98**% | **96.44**% | **77.96**% |

the original LFW dataset contains only $6,000$ pairs of faces in all, which is insufficient to evaluate deep learning methods, evidenced by the fact that results are almost saturated on this protocol. Because of this, Liao et al. [8] made use of the whole LFW dataset to build the BLUFR protocol. In this protocol, a 10-fold cross-validation test is defined for both *face verification* and *open-set face identification*. For *face verification*, a verification rate (VR) is reported for each split with strict false alarm rate (FAR= 0.1%) by comparing around $156,915$ genuine pairs and $46,960,863$ imposter pairs[1], which is closer to real-world scenario than the accuracy metric in the standard LFW protocol. For *open-set identification*, an identification rate (DIR) at Rank-1 corresponding to FAR= 1% is computed. We first test the performance of Model B without certain modules to ensure their effectiveness. Then we train the proposed Model A and Model B with all modules and compare them with base-network.

In Table 2.3, *Base-net* indicates the baseline single CNN network, which is used as the base-network in our model. *Attention Net* indicates whether an attention network is used to automatically localize the regions for sub-networks or crop the fixed regions that are randomly initialized. *Fusion Layer* indicates whether to add a fully connected fusion layer or directly use the concatenated layer as the representation. *Promotion Loss* means whether we add promotion loss as regularization to the fusion layer. The accuracy is tested on the standard protocol, while *Verification Rate* (VR) and

[1]the numbers are averaged over ten splits.

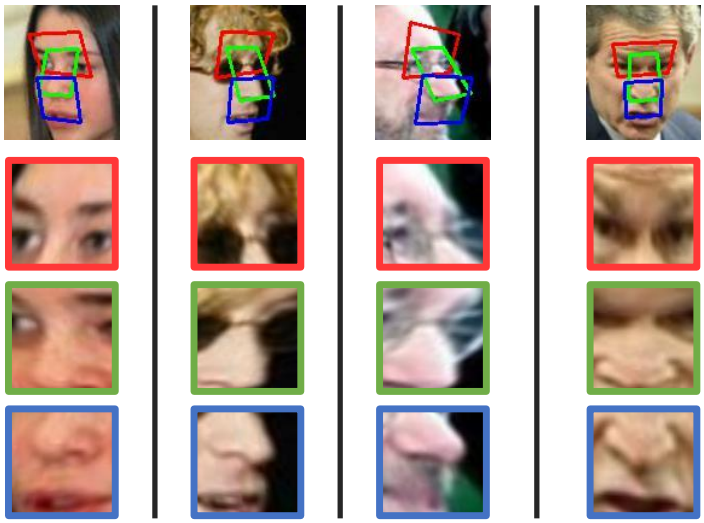


Figure 2.5 Examples of the localized regions in Model A. The attention network localizes the eyes, nose and mouth accurately by learning without landmark labels. These accurately localized patches make it an easier task for the sub-networks to learn robust features from certain facial parts.

*Detect and Identification Rate* (DIR) are tested on BLUFR protocol. Although all the results are similar on standard LFW protocol, distinct differences can be observed in BLUFR results. This is because standard protocol only contains $6,000$ pairs which is not adequate to precisely reflect the performance of a highly sophisticated model. Based on the results on BLUFR, we can see that Model B consistently outperforms base-network even without certain modules. And also every module is making a contribution and is essential to guarantee the final performance of the whole model. After using all modules, the proposed Model A and Model B surpasses the baseline by 4% in terms of DIR@FAR= 1% at rank-1. This demonstrates that the proposed idea of an auto-aligned parts-based model does improve the performance of a single neural network. And with more sub-networks added, Model B (12 sub-networks) consistently outperforms Model A (3 sub-networks).

To further evaluate the attention networks, we visualize the localized patches in Model A. Some examples are shown in Figure 2.5. Notice the different distribution of facial parts, even after alignment, due to the challenging pose of the input image. The attention network can still accurately find the target facial parts. In the localized patches in each column, all the facial parts are distributed in a similar way. These accurately localized patches make it an easier task for the sub-networks to

Table 2.4 Evaluation results on IJB-A 1:1 Comparison and 1:N Search protocols.

| | TAR@FAR (Verification) | | CMC (Closed-set Identification) | | FNIR (Open-set Identification) | |
|---|---|---|---|---|---|---|
| Type | 0.001 | 0.01 | Rank-1 | Rank-5 | 0.01 | 0.1 |
| Base-net | 0.542 ± 0.0917 | 0.7883 ± 0.0917 | 0.882 ± 0.0190 | 0.954 ± 0.0079 | 0.426 ± 0.0170 | 0.355 ± 0.0140 |
| Model A | 0.583 ± 0.0832 | 0.8075 ± 0.0264 | 0.889 ± 0.0068 | 0.957 ± 0.0068 | 0.418 ± 0.0147 | **0.353 ± 0.0137** |
| Model B | **0.602 ± 0.0692** | **0.8231 ± 0.0219** | **0.898 ± 0.0092** | **0.960 ± 0.0061** | **0.411 ± 0.0164** | 0.353 ± 0.0142 |

Table 2.5 Evaluation results on IJB-B 1:1 Baseline Verification and 1:N Mixed Media Identification protocols.

| | TAR@FAR (Verification) | | CMC (Closed-set Identification) | | FNIR (Open-set Identification) | |
|---|---|---|---|---|---|---|
| Type | 0.001 | 0.01 | Rank-1 | Rank-5 | 0.01 | 0.1 |
| Base-net | 0.631 | 0.851 | 0.749 | 0.861 | 0.149 | 0.032 |
| Model A | 0.652 | 0.861 | 0.768 | **0.875** | 0.139 | **0.031** |
| Model B | **0.659** | **0.865** | **0.769** | 0.874 | **0.135** | 0.032 |

learn robust features from certain facial parts. The attention network also allows adjusting which part to localize so that the sub-networks can find more discriminative features. Notice that the attention network is trained without the landmark labels and as such, the computation is almost free.

### 2.4.3 Evaluation on IJB-A and IJB-B Benchmarks

The IARPA Janus Benchmarks, including IJB-A and IJB-B, were released to push forward the frontiers of unconstrained face recognition systems. In IJB-A, a manually labeled dataset containing images both from photos and video frames is used to build a protocol for *face identification* (1:N Search) and *face verification* (1:1 Comparison). In comparison to LFW, the 5, 712 images and 2, 085 videos in the IJB-A benchmark have a wider geographic variation, larger pose variation and images of low resolution or heavy occlusion, making it a much harder benchmark than both standard LFW and BLUFR benchmarks. Again, a 10-fold cross-validation test is designed for both identification and verification in IJB-A. True Accept Rate (TAR) at False Accept Rate (FAR) is used to evaluate verification performance. For closed-set identification, Cumulative Match Characteristic (CMC) measures the fraction of genuine gallery templates that are retrieved within a certain rank. And False Negative Identification Rate (FNIR) at False Positive Identification Rate (FPIR) is reported to

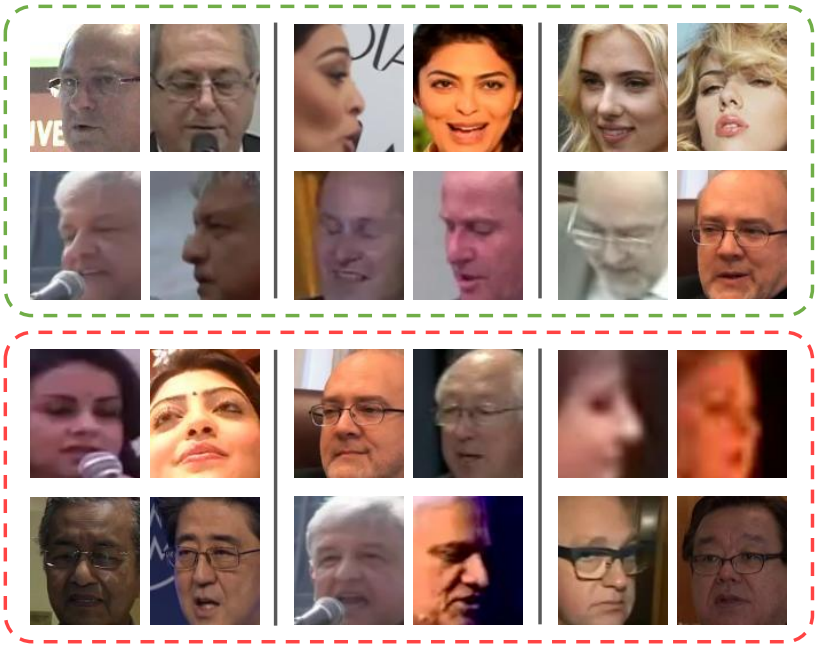


Figure 2.6 Example pairs that are misclassified by base-network but are classified correctly by Model B on IJB-B dataset. Pairs in the green box are genuine pairs and pairs in the red box are impostor pairs. We use the threshold of IJB-B 1:1 Baseline Verification for TAR@FAR= 0.1%.

evaluate the performance in terms of open-set identification.

IJB-B is an extension of IJB-A benchmark. It consists of 21,798 still images and 55,026 frames from 7,011 videos from 1,845 subjects. There is no cross-validation in IJB-B. In particular, we use the 1:1 Baseline Verification protocol and 1:N Mixed Media Identification protocol for IJB-B.

From the results in Table 2.4 and Table 2.5, we can see that the proposed models do improve the performance of the base-net on both the IJB-A and IJB-B benchmarks. This shows the effectiveness of the proposed idea which fuses features from local regions together with a global feature representation, although the base-network is already quite sophisticated. Second, Model B outperforms Model A in most protocols, which indicates that more local regions and sub-networks could help achieve even larger performance gains.

## 2.5 Conclusion

In this chapter, we have proposed a scheme for incorporating parts-based models into state-of-the-art CNNs for face recognition. A set of sub-networks are added to learn features from certain facial parts. An spatial transformer-based attention network learns to automatically localize the discriminative regions. We have further added a fusion layer to combine the global and local features, which, with the proposed promotion loss, encourages the sub-networks to find more discriminative features. The proposed approach can be applied to any single CNN to build an end-to-end system. Experiments on the most novel and challenging benchmarks show that the proposed strategy can help improve the performance of a single CNN without significant increase in run-time. Evidence suggests that we can further improve the performance with even more sub-networks.

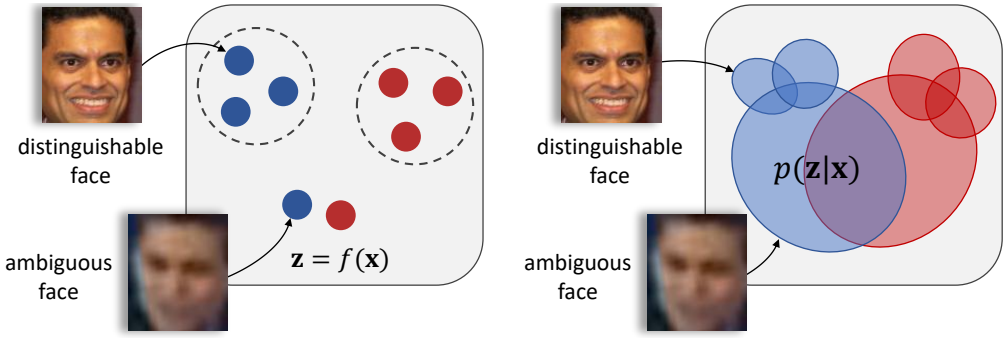
# Chapter 3

# Uncertainty Estimation for Deep Face Recognition

## 3.1 Introduction

When humans are asked to describe a face image, they not only give the description of the facial attributes, but also the confidence associated with them. For example, if the eyes are blurred in the image, a person will keep the eye size as an uncertain information and focus on other features. Furthermore, if the image is completely corrupted and no attributes can be discerned, the subject may respond that he/her cannot identify this face. This kind of uncertainty (or confidence) estimation is common and important in human decision making.

On the other hand, the representations and similarity metrics used in state-of-the-art face recognition systems are generally confidence-agnostic. These methods depend on an embedding model (e.g. Deep Neural Networks) to give a deterministic point representation for each face image in the latent feature space [30, 32, 34, 36, 38]. A point in the latent space represents the model's estimation of the facial features in the given image. If the error in the estimation is somehow bounded, the distance between two points can effectively measure the semantic similarity between the corresponding face images. But given a low-quality input, where the expected facial features are ambiguous or absent in the image, a large shift in the embedded points is inevitable, leading to false




(a) deterministic embedding (b) probabilistic embedding

Figure 3.1 Difference between deterministic face embeddings and probabilistic face embeddings (PFEs). Deterministic embeddings represent every face as a point in the latent space without regards to its feature ambiguity. Probabilistic face embedding (PFE) gives a distributional estimation of features in the latent space instead. **Best viewed in color.**

recognition (Figure 3.1a).

To address the above problems, we propose *Probabilistic Face Embeddings (PFEs)*, which give a distributional estimation instead of a point estimation in the latent space for each input face image (Figure 3.1b). The mean of the distribution can be interpreted as the most likely latent feature values while the span of the distribution represents the uncertainty of these estimations. PFE can address the unconstrained face recognition problem in a two-fold way: (1) During matching (face comparison), PFE penalizes uncertain features (dimensions) and pays more attention to more confident features. (2) For low quality inputs, the confidence estimated by PFE can be used to reject the input or actively ask for human assistance to avoid false recognition. Besides, a natural solution can be derived to aggregate the PFE representations of a set of face images into a new distribution with lower uncertainty to increase the recognition performance. The implementation of PFE is open-sourced[1]. The contributions of the chapter can be summarized as below:

1. An uncertainty-aware probabilistic face embedding (PFE) which represents face images as distributions instead of points.
2. A probabilistic framework that can be naturally derived for face matching and feature fusion using PFE.

[1] `https://github.com/seasonSH/Probabilistic-Face-Embeddings`

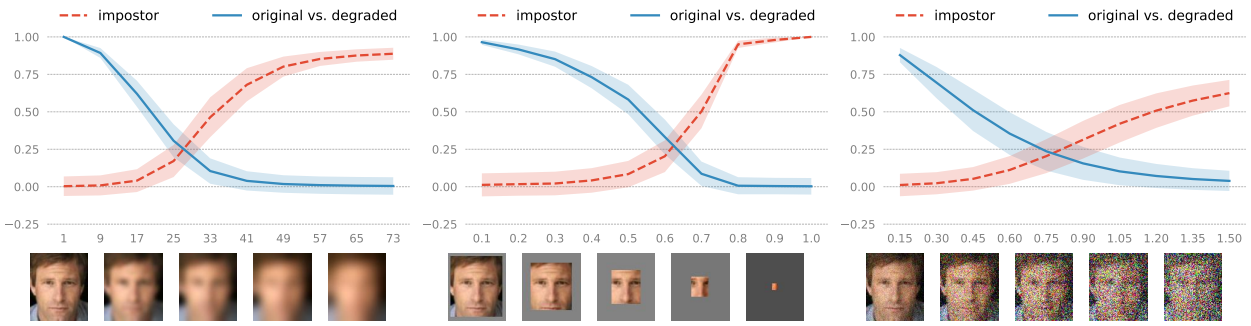
3. A simple method that converts existing deterministic embeddings into PFEs without additional training data.
4. Comprehensive experiments showing that the proposed PFE can improve face recognition performance of deterministic embeddings and can effectively filter out low-quality inputs to enhance the robustness of face recognition systems.

## 3.2 Related Work

**Uncertainty Learning in DNNs** To improve the robustness and interpretability of discriminant Deep Neural Networks (DNNs), deep uncertainty learning is getting more attention [72, 73, 74]. There are two main types of uncertainty: *model uncertainty* and *data uncertainty*. Model uncertainty refers to the uncertainty of model parameters given the training data and can be reduced by collecting additional training data [75, 76, 72, 73]. Data uncertainty accounts for the uncertainty in output whose primary source is the inherent noise in input data and hence cannot be eliminated with more training data [74]. The uncertainty studied in our work can be categorized as data uncertainty. Although techniques have been developed for estimating data uncertainty in different tasks, including classification and regression [74], they are not suitable for our task since our target space is not well-defined by given labels[2]. Variational Autoencoders [77] can also be regarded as a method for estimating data uncertainty, but it mainly serves a generation purpose. Specific to face recognition, some studies [78, 79, 80] have leveraged the model uncertainty for analysis and learning of face representations, but to our knowledge, ours is the first work that utilizes data uncertainty[3] for recognition tasks.

**Probabilistic Face Representation** Modeling faces as probabilistic distributions is not a new idea. In the field of face template/video matching, there exists abundant literature on modeling the faces as probabilistic distributions [82, 83], subspace [84] or manifolds [83, 85] in the feature space. However, the input for such methods is a set of face images rather than a single face image, and

[2]Although we are given the identity labels, they cannot directly serve as target vectors in the latent feature space.
[3]Some in the literature have also used the terminology “data uncertainty" for a different purpose [81].




(a) Gaussian Blur (b) Occlusion (c) Random Gaussian Noise

Figure 3.2 Illustration of *feature ambiguity dilemma*. The plots show the cosine similarity on LFW dataset with different degrees of degradation. Blue lines show the similarity between original images and their respective degraded versions. Red lines show the similarity between impostor pairs of degraded images. The shading indicates the standard deviation. With larger degrees of degradation, the model becomes more confident (very high/low scores) in a wrong way.

they use a between-distribution similarity or distance measure, e.g. KL-divergence, for comparison, which does not penalize the uncertainty. Meanwhile, some studies [86, 87] have attempted to build a fuzzy model of a given face using the features of face parts. In comparison, the proposed PFE represents each single face image as a distribution in the latent space encoded by DNNs and we use an uncertainty-aware log likelihood score to compare the distributions.

**Quality-aware Pooling** In contrast to the methods above, recent work on face template/video matching aims to leverage the saliency of deep CNN embeddings by aggregating the deep features of all faces into a single compact vector [88, 89, 90, 91]. In these methods, a separate module learns to predict the quality of each face in the image set, which is then normalized for a weighted pooling of feature vectors. We show that a solution can be naturally derived under our framework, which not only gives a probabilistic explanation for quality-aware pooling methods, but also leads to a more general solution where an image set can also be modeled as a PFE representation.

## 3.3 Limitations of Deterministic Embeddings

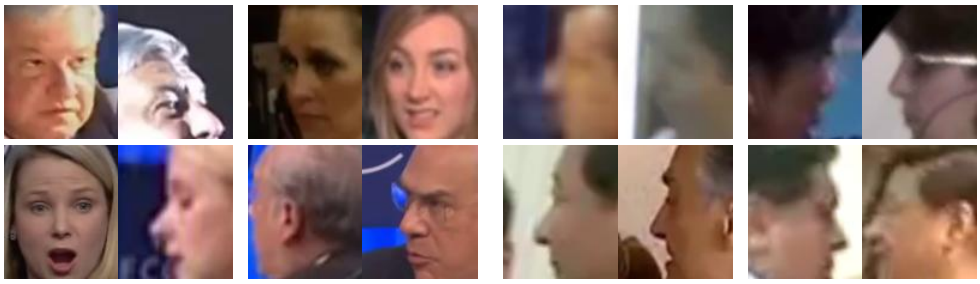
In this section, we explain the problems of deterministic face embeddings from both theoretical and empirical views. Let $\mathcal{X}$ denote the image space and $\mathcal{Z}$ denote the latent feature space of $D$

dimensions. An ideal latent space $\mathcal{Z}$ should only encode *identity-salient* features and be *disentangled* from identity-irrelevant features. As such, each identity should have a unique intrinsic code $\mathbf{z} \in \mathcal{Z}$ that best represents this person and each face image $\mathbf{x} \in \mathcal{X}$ is an observation sampled from $p(\mathbf{x}|\mathbf{z})$. The process of training face embeddings can be viewed as a joint process of searching for such a latent space $\mathcal{Z}$ and learning the inverse mapping $p(\mathbf{z}|\mathbf{x})$. For deterministic embeddings, the inverse mapping is a Dirac delta function $p(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - f(\mathbf{x}))$, where $f$ is the embedding function. Clearly, for any space $\mathcal{Z}$, given the possibility of noises in $\mathbf{x}$, it is unrealistic to recover the exact $\mathbf{z}$ and the embedded point of a low-quality input would inevitably shift away from its intrinsic $\mathbf{z}$ (no matter how much training data we have).

The question is whether this shift could be bounded such that we still have smaller intra-class distances compared to inter-class distances. However, this is unrealistic for fully unconstrained face recognition and we conduct an experiment to illustrate this. Let us start with a simple example: given a pair of identical images, a deterministic embedding will always map them to the same point and therefore the distance between them will always be 0, even if these images do not contain a face. This implies that "a pair of images being similar or even the same does not necessarily mean the probability of their belonging to the same person is high".

To demonstrate this, we conduct an experiment by manually degrading the high-quality images and visualizing their similarity scores. We randomly select a high-quality image of each subject from the LFW dataset [3] and manually insert Gaussian blur, occlusion, and random Gaussian noise to the faces. In particular, we linearly increase the size of Gaussian kernel, occlusion ratio and the standard deviation of the noise to control the degradation degree. At each degradation level, we extract the feature vectors with a 64-layer CNN[4], which is comparable to state-of-the-art face recognition systems. The features are normalized to a hyper-spherical embedding space. Then, two types of cosine similarities are reported: (1) similarity between pairs of original image and its respective degraded image, and (2) similarity between degraded images of different identities. As shown in Figure 3.2, for all the three types of degradation, the genuine similarity scores decrease to

[4]trained on Ms-Celeb-1M [2] with AM-Softmax [35]



(a) Low-similarity Genuine Pairs (b) High-similarity Impostor Pairs

Figure 3.3 Example genuine pairs from IJB-A dataset estimated with the lowest similarity scores and impostor pairs with the highest similarity scores (among all possible pairs) by a 64-layer CNN model. The genuine pairs mostly consist of one high-quality and one low-quality image while the impostor pairs are all low-quality images. Note that these pairs are not templates in the verification protocol.

0 while the impostor similarity scores converge to 1.0! These indicate two types of errors that can be expected in a fully unconstrained scenario even when the model is very confident (very high/low similarity scores):

(1) false accept of impostor low-quality pairs and

(2) false reject of genuine cross-quality pairs.

To confirm this, we test the model on the IJB-A dataset by finding impostor/genuine image pairs with the highest/lowest scores, respectively. The situation is exactly as we hypothesized (See Figure 3.3). We call this *Feature Ambiguity Dilemma* which is observed when the deterministic embeddings are forced to estimate the features of ambiguous faces. The experiment also implies that there exist a *dark space* where the ambiguous inputs are mapped to and the distance metric is distorted.

## 3.4 Probabilistic Face Embeddings

To address the aforementioned problem caused by data uncertainty, we propose to encode the uncertainty into the face representation and take it into account during matching. Specifically, instead of building a model that gives a point estimation in the latent space, we estimate a distribution $p(\mathbf{z}|\mathbf{x})$ in the latent space to represent the potential appearance of a person's face[5]. In particular, we

[5]following the notations in Section 3.3.

use a multivariate Gaussian distribution:

$$p(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \mathbf{I}) \tag{3.1}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are both a $D$-dimensional vector predicted by the network from the $i^{\text{th}}$ input image $\mathbf{x}_i$. Here we only consider a diagonal covariance matrix to reduce the complexity of the face representation. This representation should have the following properties:

1. The center $\boldsymbol{\mu}$ should encode the most likely facial features of the input image.
2. The uncertainty $\boldsymbol{\sigma}$ should encode the model's confidence along each feature dimension.

In addition, we wish to use a single network to predict the distribution. Considering that new approaches for training face embeddings are still being developed, we aim to develop a method that could convert existing deterministic face embedding networks to PFEs in an easy manner. In the followings, we first show how to compare and fuse the PFE representations to demonstrate their strength and then propose our method for learning PFEs.

### 3.4.1 Matching with PFEs

Given the PFE representations of a pair of images $(\mathbf{x}_i, \mathbf{x}_j)$, we can directly measure the "likelihood" of them belonging to the same person (sharing the same latent code): $p(\mathbf{z}_i = \mathbf{z}_j)$, where $\mathbf{z}_i \sim p(\mathbf{z}|\mathbf{x}_i)$ and $\mathbf{z}_j \sim p(\mathbf{z}|\mathbf{x}_j)$. Specifically,

$$p(\mathbf{z}_i = \mathbf{z}_j) = \int p(\mathbf{z}_i|\mathbf{x}_i) p(\mathbf{z}_j|\mathbf{x}_j) \delta(\mathbf{z}_i - \mathbf{z}_j) d\mathbf{z}_i d\mathbf{z}_j. \tag{3.2}$$

In practice, we would like to use the log likelihood instead, whose solution is given by:

$$\begin{aligned} s(\mathbf{x}_i, \mathbf{x}_j) &= \log p(\mathbf{z}_i = \mathbf{z}_j) \\ &= -\frac{1}{2} \sum_{l=1}^{D} \left( \frac{(\mu_i^{(l)} - \mu_j^{(l)})^2}{\sigma_i^{2(l)} + \sigma_j^{2(l)}} + \log(\sigma_i^{2(l)} + \sigma_j^{2(l)}) \right) \\ &\quad - const, \end{aligned} \tag{3.3}$$

where $const = \frac{D}{2}\log 2\pi$, $\mu_i^{(l)}$ refers to the $l^{\text{th}}$ dimension of $\boldsymbol{\mu}_i$ and similarly for $\sigma_i^{(l)}$.

Note that this symmetric measure can be viewed as the expectation of likelihood of one input's latent code conditioned on the other, that is

$$\begin{aligned} s(\mathbf{x}_i, \mathbf{x}_j) &= \log \int p(\mathbf{z}|\mathbf{x}_i) p(\mathbf{z}|\mathbf{x}_j) d\mathbf{z} \\ &= \log \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}_i)}[p(\mathbf{z}|\mathbf{x}_j)] \\ &= \log \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}_j)}[p(\mathbf{z}|\mathbf{x}_i)]. \end{aligned} \tag{3.4}$$

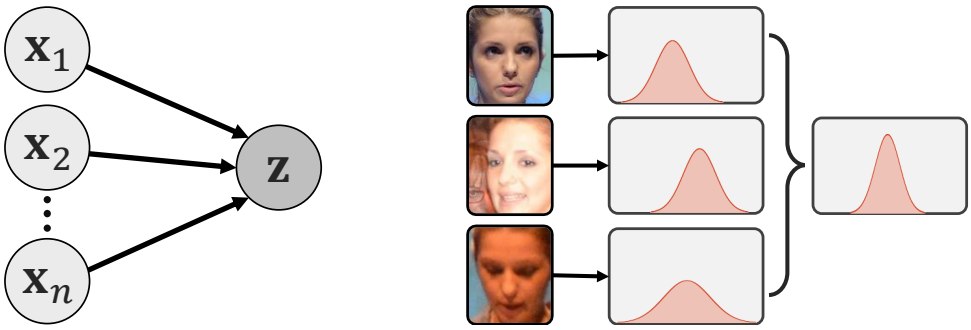
As such, we call it *mutual likelihood score (MLS)*. Different from KL-divergence, this score is unbounded and cannot be seen as a distance metric. It can be shown that the squared Euclidean distance is equivalent to a special case of MLS when all the uncertainties are assumed to be the same:

**Property 1** *If $\sigma_i^{(l)}$ is a fixed number for all data $\mathbf{x}_i$ and dimensions l, MLS is equivalent to a scaled and shifted negative squared Euclidean distance.*

Further, when the uncertainties are allowed to be different, we note that MLS has some interesting properties that make it different from a distance metric:

1. *Attention* mechanism: the first term in the bracket in Equation (3.3) can be seen as a weighted distance which assigns larger weights to less uncertain dimensions.
2. *Penalty* mechanism: the second term in the bracket in Equation (3.3) can be seen as a penalty term which penalizes dimensions that have high uncertainties.
3. If either input $\mathbf{x}_i$ or $\mathbf{x}_j$ has large uncertainties, MLS will be low (because of penalty) irrespective of the distance between their mean.
4. Only if both inputs have small uncertainties and their means are close to each other, MLS could be very high.

The last two properties imply that PFE could solve the feature ambiguity dilemma if the network can effectively estimate $\boldsymbol{\sigma}_i$.




(a) (b)

Figure 3.4 Fusion with PFEs. (a) Illustration of the fusion process as a directed graphical model. (b) Given the Gaussian representations of faces (from the same identity), the fusion process outputs a new Gaussian distribution in the latent space with a more precise mean and lower uncertainty.

### 3.4.2 Fusion with PFEs

In many cases we have a template (set) of face images, for which we need to build a compact representation for matching. With PFEs, a conjugate formula can be derived for representation fusion (Figure 3.4). Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a series of observations (face images) from the same identity and $p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be the posterior distribution after the $n^{\text{th}}$ observation. Then, assuming all the observations are conditionally independent (given the latent code $\mathbf{z}$). It can be shown that:

$$p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n+1}) = \alpha \frac{p(\mathbf{z}|\mathbf{x}_{n+1})}{p(\mathbf{z})} p(\mathbf{z}|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \tag{3.5}$$

where $\alpha$ is a normalization factor. To simplify the notations, let us only consider a one-dimensional case below; the solution can be easily extended to the multivariate case.

If $p(\mathbf{z})$ is assumed to be a noninformative prior,i.e. $p(\mathbf{z})$ is a Gaussian distribution whose variance approaches $\infty$, the posterior distribution in Equation (3.5) is a new Gaussian distribution with lower uncertainty.

Further, given a set of face images $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the parameters of the fused representation

can be directly given by:

$$\hat{\mu}_n = \sum_{i=1}^{n} \frac{\hat{\sigma}_n^2}{\sigma_i^2} \mu_i, \tag{3.6}$$

$$\frac{1}{\hat{\sigma}_n^2} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2}. \tag{3.7}$$

In practice, because the conditional independence assumption is usually not true, e.g. video frames include a large amount of redundancy, Equation ( 3.7) will be biased by the number of images in the set. Therefore, we take dimension-wise minimum to obtain the new uncertainty.

**Relationship to Quality-aware Pooling** If we consider a case where all the dimensions share the same uncertainty $\sigma_i$ for $i^{\text{th}}$ input and let the quality value $q_i = \frac{1}{\sigma_i^2}$ be the output of the network. Then Equation (3.6) can be written as

$$\hat{\boldsymbol{\mu}}_n = \frac{\sum_{i=1}^{n} q_i \boldsymbol{\mu}_i}{\sum_j^n q_j}. \tag{3.8}$$

If we do not use the uncertainty after fusion, the algorithm will be the same as recent quality-aware aggregation methods for set-to-set face recognition [88, 89, 90].

### 3.4.3 Learning

Note that any deterministic embedding $f$, if properly optimized, can indeed satisfy the properties of PFEs: (1) the embedding space is a disentangled identity-salient latent space and (2) $f(\mathbf{x})$ represents the most likely features of the given input in the latent space. As such, in this work we consider a stage-wise training strategy: given a pre-trained embedding model $f$, we fix its parameters, take $\boldsymbol{\mu}(\mathbf{x}) = f(\mathbf{x})$, and optimize an additional uncertainty module to estimate $\boldsymbol{\sigma}(\mathbf{x})$. When the uncertainty module is trained on the same dataset of the embedding model, this stage-wise training strategy allows us to have a more fair comparison between PFE and the original embedding $f(\mathbf{x})$ than an end-to-end learning strategy.

The uncertainty module is a network with two fully-connected layers which shares the same

input as of the bottleneck layer[6]. The optimization criteria is to maximize the mutual likelihood score of all genuine pairs $(\mathbf{x}_i, \mathbf{x}_j)$. Formally, the loss function to minimize is

$$\mathcal{L} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} -s(\mathbf{x}_i, \mathbf{x}_j) \tag{3.9}$$

where $\mathcal{P}$ is the set of all genuine pairs and $s$ is defined in Equation (3.3). In practice, the loss function is optimized within each mini-batch. Intuitively, this loss function can be understood as an alternative to maximizing $p(\mathbf{z}|\mathbf{x})$: if the latent distributions of all possible genuine pairs have a large overlap, the latent target $\mathbf{z}$ should have a large likelihood $p(\mathbf{z}|\mathbf{x})$ for any corresponding $\mathbf{x}$. Notice that because $\boldsymbol{\mu}(\mathbf{x})$ is fixed, the optimization wouldn't lead to the collapse of all the $\boldsymbol{\mu}(\mathbf{x})$ to a single point.

## 3.5 Implementation Details

All the models in the chapter are implemented using Tensorflow r1.9. Two and Four GeForce GTX 1080 Ti GPUs are used for training base models on CASIA-Webface [50] and MS-Celeb-1M [2], respectively. The uncertainty modules are trained using one GPU.

### 3.5.1 Data Preprocessing

All the face images are first passed through MTCNN face detector [7] to detect 5 facial landmarks (two eyes, nose and two mouth corners). Then, similarity transformation is used to normalize the face images based on the five landmarks. After transformation, the images are resized to $112 \times 96$. Before passing into networks, each pixel in the RGB image is normalized by subtracting 127.5 and dividing by 128.

[6]Bottleneck layer refers to the layer which outputs the original face embedding.

### 3.5.2 Base Models

The base models for CASIA-Webface [50] are trained for $28,000$ steps using a SGD optimizer with a momentum of 0.9. The learning rate starts at 0.1, and is decreased to 0.01 and 0.001 after $16,000$ and $24,000$ steps, respectively. For the base model trained on Ms-Celeb-1M [2], we train the network for $140,000$ steps using the same optimizer settings. The learning rate starts at 0.1, and is decreased to 0.01 and 0.001 after $80,000$ and $120,000$ steps, respectively. The batch size, feature dimension and weight decay are set to 256, 512 and 0.0005, respectively, for both cases.

### 3.5.3 Uncertainty Module

**Architecture** The uncertainty module for all models are two-layer perceptrons with the same architecture: `FC-BN-ReLU-FC-BN-exp`, where `FC` refers to fully connected layers, `BN` refers to batch normalization layers [69] and `exp` function ensures the outputs $\sigma^2$ are all positive values [74]. The first `FC` shares the same input with the bottleneck layer, i.e. the output feature map of the last convolutional layer. The output of both `FC` layers are $D$-dimensional vectors, where $D$ is the dimensionality of the latent space. In addition, we constrain the last `BN` layer to share the same $\gamma$ and $\beta$ across all dimensions, which we found to help stabilizing the training.

**Training** For the models trained on CASIA-WebFace [50], we train the uncertainty module for $3,000$ steps using a SGD optimizer with a momentum of 0.9. The learning rate starts at 0.001, and is decreased to 0.0001 after $2,000$ steps. For the model trained on MS-Celeb-1M[2], we train the uncertainty module for $12,000$ steps. The learning rate starts at 0.001, and is decreased to 0.0001 after $8,000$ steps. The batch size for both cases are 256. For each mini-batch, we randomly select 4 images from 64 different subjects to compose the positive pairs (384 pairs in all). The weight decay is set to 0.0005 in all cases. A Subset of the training data was separated as the validation set for choosing these hyper-parameters during development phase.

Table 3.1 Results of models trained on CASIA-WebFace. "Original" refers to the deterministic embeddings. The better performance among each base model are shown in bold numbers. "PFE" uses mutual likelihood score for matching. IJB-A results are verification rates at FAR=0.1%.

| Base Model | Representation | LFW | YTF | CFP-FP | IJB-A |
|---|---|---|---|---|---|
| Softmax + Center Loss [32] | Original | 98.93 | 94.74 | 93.84 | 78.16 |
| | PFE | **99.27** | **95.42** | **94.51** | **80.83** |
| Triplet [30] | Original | 97.65 | 93.36 | 89.76 | 60.82 |
| | PFE | **98.45** | **93.96** | **90.04** | **61.00** |
| A-Softmax [34] | Original | 99.15 | 94.80 | 92.41 | 78.54 |
| | PFE | **99.32** | **94.94** | **93.37** | **82.58** |
| AM-Softmax [35] | Original | 99.28 | 95.64 | 94.77 | 84.69 |
| | PFE | **99.55** | **95.92** | **95.06** | **87.58** |

**Inference Speed** Feature extraction (passing through the whole network) using one GPU takes 1.5ms per image. Note that given the small size of the uncertainty module, it has little impact on the feature extraction time. Matching images using cosine similarity and mutual likelihood score takes 4ns and 15ns , respectively. Both are neglectable in comparison with feature extraction time.

## 3.6 Experiments

In this section, we first test the proposed PFE method on standard face recognition protocols to compare with deterministic embeddings. Then we conduct qualitative analysis to gain more insight into how PFE behaves.

To comprehensively evaluate the efficacy of PFEs, we conduct the experiments on 7 benchmarks, including the well known **LFW** [3], **YTF** [51], **MegaFace** [52] and four other more unconstrained benchmarks.

We use the CASIA-WebFace [50] and a cleaned version[7] of MS-Celeb-1M [2] as training data, from which we remove the subjects that are also included in the test datasets[8].

[7] `https://github.com/inlmouse/MS-Celeb-1M_WashList`.

[8] 84 and 4, 182 subjects were removed from CASIA-WebFace and MS-Celeb-1M, respectively.

Table 3.2 Results of our models (last three rows) trained on MS-Celeb-1M and state-of-the-art methods on LFW, YTF and MegaFace. The MegaFace verification rates are computed at FAR=0.0001%. "-" indicates that the author did report the performance on the corresponding protocol.

| Method | Training Data | LFW | YTF | MF1 Rank1 | MF1 Veri. |
|---|---|---|---|---|---|
| DeepFace+ [27] | 4M | 97.35 | 91.4 | - | - |
| FaceNet [30] | 200M | 99.63 | 95.1 | - | - |
| DeepID2+ [31] | 300K | 99.47 | 93.2 | - | - |
| CenterFace [32] | 0.7M | 99.28 | 94.9 | 65.23 | 76.52 |
| SphereFace [34] | 0.5M | 99.42 | 95.0 | 75.77 | 89.14 |
| ArcFace [38] | 5.8M | 99.83 | 98.02 | 81.03 | 96.98 |
| CosFace [36] | 5M | 99.73 | 97.6 | 77.11 | 89.88 |
| L2-Face [37] | 3.7M | 99.78 | 96.08 | - | - |
| Baseline | 4.4M | 99.70 | 97.18 | 79.43 | 92.93 |
| $\text{PFE}_{\text{fuse}}$ | 4.4M | - | 97.32 | - | - |
| $\text{PFE}_{\text{fuse+match}}$ | 4.4M | 99.82 | 97.36 | 78.95 | 92.51 |

### 3.6.1 Experiments on Different Base Embeddings

Since our method works by converting existing deterministic embeddings, we want to evaluate how it works with different base embeddings, i.e. face representations trained with different loss functions. In particular, we implement the following state-of-the-art loss functions: Softmax+Center Loss [32], Triplet Loss [30], A-Softmax [34] and AM-Softmax [35][9]. To be aligned with previous work [34, 36], we train a 64-layer residual network [34] with each of these loss functions on the CASIA-WebFace dataset as base models. All the features are $\ell 2$-normalized to a hyper-spherical embedding space. Then we train the uncertainty module for each base model on the CASIA-WebFace again for $3,000$ steps. We evaluate the performance on four benchmarks: LFW [3], YTF [51], CFP-FP [4] and IJB-A [5], which present different challenges in face recognition. The results are shown in Table 3.1. The PFE improves over the original representation in all cases, indicating the proposed method is robust with different embeddings and testing scenarios.

Table 3.3 Results of our models (last three rows) trained on MS-Celeb-1M and state-of-the-art methods on CFP (frontal-profile protocol) and IJB-A.

| Method | Training Data | IJB-A (TAR@FAR) 0.1% | IJB-A (TAR@FAR) 1.0% | CFP-FP |
|---|---|---|---|---|
| Yin et.al. [92] | 0.5M | 73.9 ± 4.2 | 77.5 ± 2.5 | **94.39** |
| NAN [88] | 3M | 88.1 ± 1.1 | 94.1 ± 0.8 | - |
| QAN [89] | 5M | 89.31 ± 3.92 | 94.20 ± 1.53 | - |
| Cao et.al. [44] | 3.3M | 90.4 ± 1.4 | 95.8 ± 0.6 | - |
| Multicolumn [90] | 3.3M | 92.0 ± 1.3 | 96.2 ± 0.5 | - |
| L2-Face [37] | 3.7M | 94.3 ± 0.5 | 97.00 ± 0.4 | - |
| Baseline | 4.4M | 93.30 ± 1.29 | 96.15 ± 0.71 | 92.78 |
| $PFE_{fuse}$ | 4.4M | 94.59 ± 0.72 | 95.92 ± 0.73 | - |
| $PFE_{fuse+match}$ | 4.4M | **95.25 ± 0.89** | **97.50 ± 0.43** | 93.34 |

Table 3.4 Results of our models (last three rows) trained on MS-Celeb-1M and state-of-the-art methods on IJB-C.

| Method | Training Data | IJB-C (TAR@FAR) 0.001% | 0.01% | 0.1% | 1% |
|---|---|---|---|---|---|
| Yin et.al. [93] | 0.5M | - | - | 69.3 | 83.8 |
| Cao et.al. [44] | 3.3M | 74.7 | 84.0 | 91.0 | 96.0 |
| Multicolumn [90] | 3.3M | 77.1 | 86.2 | 92.7 | 96.8 |
| DCN [94] | 3.3M | - | 88.5 | 94.7 | **98.3** |
| Baseline | 4.4M | 70.10 | 85.37 | 93.61 | 96.91 |
| $PFE_{fuse}$ | 4.4M | 83.14 | 92.38 | 95.47 | 97.36 |
| $PFE_{fuse+match}$ | 4.4M | **89.64** | **93.25** | **95.49** | 97.17 |

Table 3.5 Performance comparison on three protocols of IJB-S. The performance is reported in terms of rank retrieval (closed-set) and TPIR@FPIR (open-set) instead of the media-normalized version [1]. The numbers "1%" and "10%" in the second row refer to the FPIR.

| Method | Training Data | Surveillance-to-Single | | | | | Surveillance-to-Booking | | | | | Surveillance-to-Surveillance | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Rank-1 | Rank-5 | Rank-10 | 1% | 10% | Rank-1 | Rank-5 | Rank-10 | 1% | 10% | Rank-1 | Rank-5 | Rank-10 | 1% | 10% |
| C-FAN [91] | 5.0M | 50.82 | 61.16 | 64.95 | 16.44 | 24.19 | 53.04 | 62.67 | 66.35 | 27.40 | 29.70 | **10.05** | 17.55 | 21.06 | 0.11 | 0.68 |
| Baseline | 4.4M | 50.00 | 59.07 | 62.70 | 7.22 | 19.05 | 47.54 | 56.14 | 61.08 | 14.75 | 22.99 | 9.40 | 17.52 | 23.04 | 0.06 | 0.71 |
| $PFE_{fuse}$ | 4.4M | **53.44** | **61.40** | **65.05** | 10.53 | 22.87 | **55.45** | **63.17** | **66.38** | 16.70 | 26.20 | 8.18 | 14.52 | 19.31 | 0.09 | 0.63 |
| $PFE_{fuse+match}$ | 4.4M | 50.16 | 58.33 | 62.28 | **31.88** | **35.33** | 53.60 | 61.75 | 64.97 | **35.99** | **39.82** | 9.20 | **20.82** | **27.34** | **0.84** | **2.83** |

### 3.6.2 Comparison with State-Of-The-Art

To compare with state-of-the-art face recognition methods, we use a different base model, which is a 64-layer network trained with AM-Softmax on the MS-Celeb-1M dataset. Then we fix the parameters and train the uncertainty module on the same dataset for $12,000$ steps. In the following experiments, we compare 3 methods:

- **Baseline** only uses the original features of the 64-layer deterministic embedding along with cosine

[9]We also tried implementing ArcFace [38] but it does not converge well in our case. So we did not use it.

similarity for matching. Average pooling is used in case of template/video benchmarks.

- **PFE$_{\text{fuse}}$** uses the uncertainty estimation $\sigma$ in PFE and Equation (3.6) to aggregate the features of templates but uses cosine similarity for matching. If the uncertainty module could estimate the feature uncertainty effectively, fusion with $\sigma$ should be able to outperform average pooling by assigning larger weights to confident features.
- **PFE$_{\text{fuse+match}}$** uses $\sigma$ both for fusion and matching (with mutual likelihood scores). Templates/videos are fused based on Equation (3.6) and Equation (3.7).

In Table 3.2 we show the results on three relatively easier benchmarks: LFW, YTF and MegaFace. Although the accuracy on LFW and YTF are nearly saturated, the proposed PFE still improves the performance of the original representation. Note that MegaFace is a biased dataset: because all the probes are high-quality images from FaceScrub, the positive pairs in MegaFace are both high-quality images while the negative pairs only contain at most one low-quality image[10] . Therefore, neither of the two types of error caused by the feature ambiguity dilemma (Section 3.3) will show up in MegaFace and it naturally favors deterministic embeddings. However, the PFE still maintains the performance in this case. We also note that such a bias, namely the target gallery images being of higher quality than the rest of gallery, would not exist in real world applications.

In Table 3.3 and Table 3.4 we show the results on three more challenging datasets: CFP, IJB-A and IJB-C. The images in these datasets present larger variations in pose, occlustion, etc, and facial features could be more ambiguous. As such, we can see that PFE achieves a more significant improvement on these three benchmarks. In particular on IJB-C at FAR= 0.001%, PFE reduces the error rate by 64%. In addition, simply fusing the original features with the learned uncertainty (PFE$_{\text{fuse}}$) also helps the performance.

In Table 3.5 we report the results on three protocols of the latest benchmark, IJB-S. Again, PFE is able to improve the performance in most cases. Notice that the gallery templates in the "Surveillance-to-still" and "Surveillance-to-booking" all include high-quality frontal mugshots, which present little feature ambiguity. Therefore, we only see a slight performance gap in these two

[10]The negative pairs of MegaFace in the verification protocol only include those between probes and distractors.

Table 3.6 Results of different network architectures trained on CASIA-WebFace. "Original" refers to the deterministic embeddings. The better performance among each base model are shown in bold numbers. "PFE" uses mutual likelihood score for matching. IJB-A results are verification rates at FAR=0.1%.

(a) CASIA-Net

| Base Model | Representation | LFW | YTF | CFP-FP | IJB-A |
|---|---|---|---|---|---|
| Softmax + Center Loss [32] | Original | 97.70 | 92.56 | 91.13 | 63.93 |
| | PFE | **97.89** | **93.10** | **91.36** | **64.33** |
| Triplet [30] | Original | 96.98 | 90.72 | **85.69** | **54.47** |
| | PFE | **97.10** | **91.22** | 85.10 | 51.35 |
| A-Softmax [34] | Original | 97.12 | **92.38** | 89.31 | 54.48 |
| | PFE | **97.92** | 91.78 | **89.96** | **58.09** |
| AM-Softmax [35] | Original | 98.32 | 93.50 | 90.24 | 71.28 |
| | PFE | **98.63** | **94.00** | **90.50** | **75.92** |

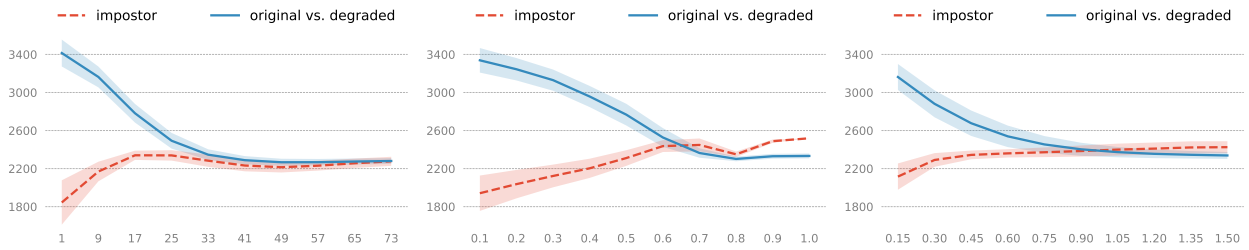
(b) Light-CNN

| Base Model | Representation | LFW | YTF | CFP-FP | IJB-A |
|---|---|---|---|---|---|
| Softmax + Center Loss [32] | Original | 97.77 | 92.34 | 90.96 | 60.42 |
| | PFE | **98.28** | **93.24** | **92.29** | **62.41** |
| Triplet [30] | Original | 97.48 | 92.46 | 90.01 | 52.34 |
| | PFE | **98.15** | **93.62** | **90.54** | **56.81** |
| A-Softmax [34] | Original | 98.07 | 92.72 | 89.34 | 63.21 |
| | PFE | **98.47** | **93.44** | **90.54** | **71.96** |
| AM-Softmax [35] | Original | 98.68 | 93.78 | 90.59 | 76.50 |
| | PFE | **98.95** | **94.34** | **91.26** | **80.00** |

protocols. But in the most challenging "surveillance-to-surveillance" protocol, larger improvement can be achieved by using uncertainty for matching. Besides, $\text{PFE}_{\text{fuse+match}}$ improves the performance significantly on all the open-set protocols, which indicates that MLS has more impact on the absolute pairwise score than the relative ranking.

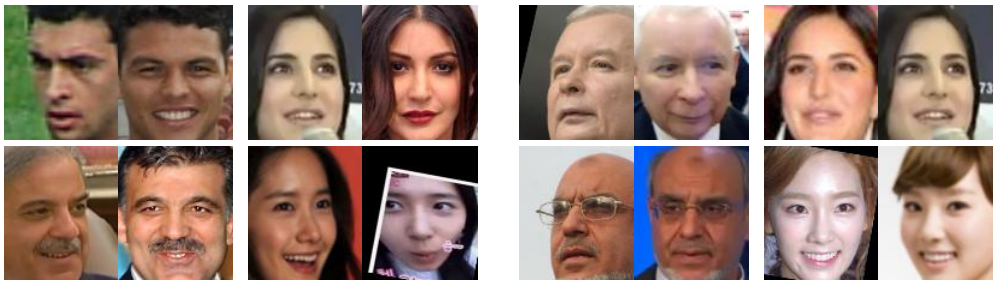
## 3.7 Results on Different Architectures

Here, we evaluate the proposed method on two different network architectures for face recognition: CASIA-Net [50] and 29-layer Light-CNN [95]. Notice that both networks require different image shapes from our preprocessed ones. Thus we pad our images with zero values and resize them into the target size. Since the main purpose of the experiment is to evaluate the efficacy of the uncertainty module rather than comparing with the original results of these networks, the difference in the preprocessing should not affect a fair comparison. Besides, the original CASIA-Net does not converge for A-Softmax and AM-Softmax, so we add an bottleneck layer to output the embedding representation after the average pooling layer. Then we conduct the experiments by comparing probabilistic embeddings with base deterministic embeddings, similar to Section 3.6.1 . The results are shown in Table 3.6a and Table 3.6b. Without tuning the architecture of the uncertainty module nor the hyper-parameters, PFE still improve the performance in most cases.




(a) Gaussian Blur (b) Occlusion (c) Random Noise

Figure 3.5 Repeated experiments on feature ambiguity dilemma with the proposed PFE. The same model in Figure 3.2 is used as the base model and is converted to a PFE by training an uncertainty module. No additional training data nor data augmentation is used for training.



(a) Low-score Genuine Pairs (b) High-score Impostor Pairs

Figure 3.6 Example genuine pairs from IJB-A dataset estimated with the lowest mutual likelihood scores and impostor pairs with the highest scores by the PFE version of the same 64-layer CNN model in Section 3.3. In comparison to Figure 3.3, most images here are high-quality ones with clear features, which can mislead the model to be confident in a wrong way. Note that these pairs are not templates in the verification protocol.

### 3.7.1 Qualitative Analysis

**Why and when does PFE improve performance?** We first repeat the same experiments in Section 3.3 using the PFE representation and MLS. The same network is used as the base model here. As one can see in Figure 3.5, although the scores of low-quality impostor pairs are still increasing, they converge to a point that is lower than the majority of genuine scores. Similarly, the scores of cross-quality genuine pairs converge to a point that is higher than the majority of impostor scores. This means the two types of errors discussed in Section 3.3 could be solved by PFE. This is further confirmed by the IJB-A results in Figure 3.6. Figure 3.7 shows the distribution of estimated uncertainty on LFW, IJB-A and IJB-S. As one can see, the "variance" of uncertainty increases in the following order: LFW < IJB-A < IJB-S. Comparing with the performance in Section 3.6.2, we can

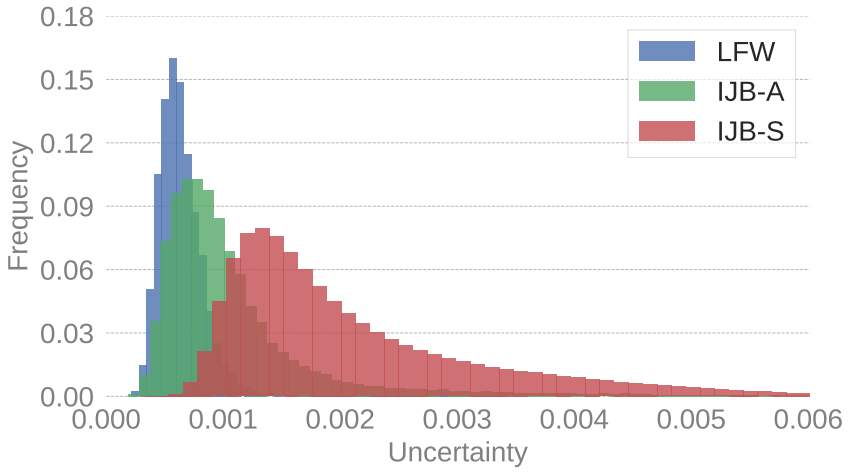



Figure 3.7 Distribution of estimated uncertainty on different datasets. Here, "Uncertainty" refers to the harmonic mean of $\sigma$ across all feature dimensions. Note that the estimated uncertainty is proportional to the complexity of the datasets. **Best viewed in color**.

see that PFE tends to achieve larger performance improvement on datasets with more diverse image quality.

**What does DNN see and not see?** To answer this question, we train a decoder network on the original embedding, then apply it to PFE by sampling $\mathbf{z}$ from the estimated distribution $p(\mathbf{z}|\mathbf{x})$ of given $\mathbf{x}$. For a high-quality image (Figure 3.8 Row 1), the reconstructed images tend to be very consistent without much variation, implying the model is very certain about the facial features in this images. In contrast, for a lower-quality input (Figure 3.8 Row 2), larger variation can be observed from the reconstructed images. In particular, attributes that can be clearly discerned from the image (e.g. thick eye-brow) are still consistent while attributes cannot (e.g. eye shape) be discerned have larger variation. As for a mis-detected image (Figure 3.8 Row 3), significant variation can be observed in the reconstructed images: the model does not see any salient feature in the given image.

## 3.8 Risk-controlled Face Recognition

In many scenarios, we may expect a higher performance than our system is able to achieve or we may want to make sure the system's performance can be controlled when facing complex application scenarios. Therefore, we would expect the model to reject input images if it is not confident. A

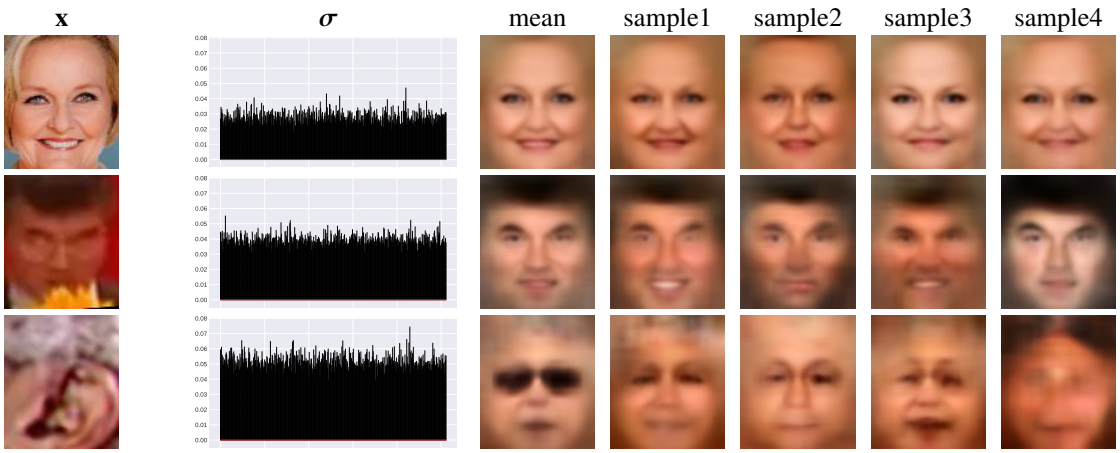
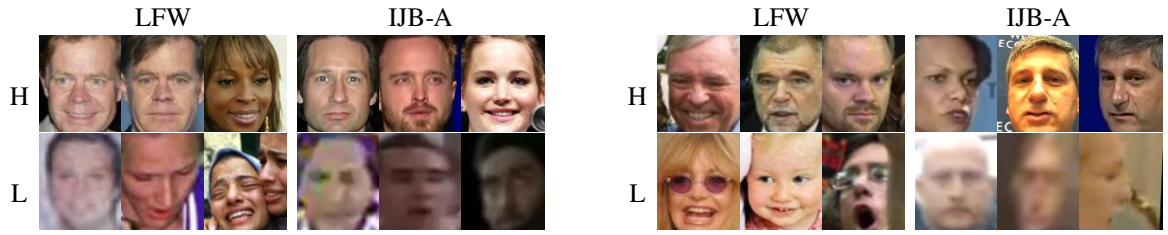


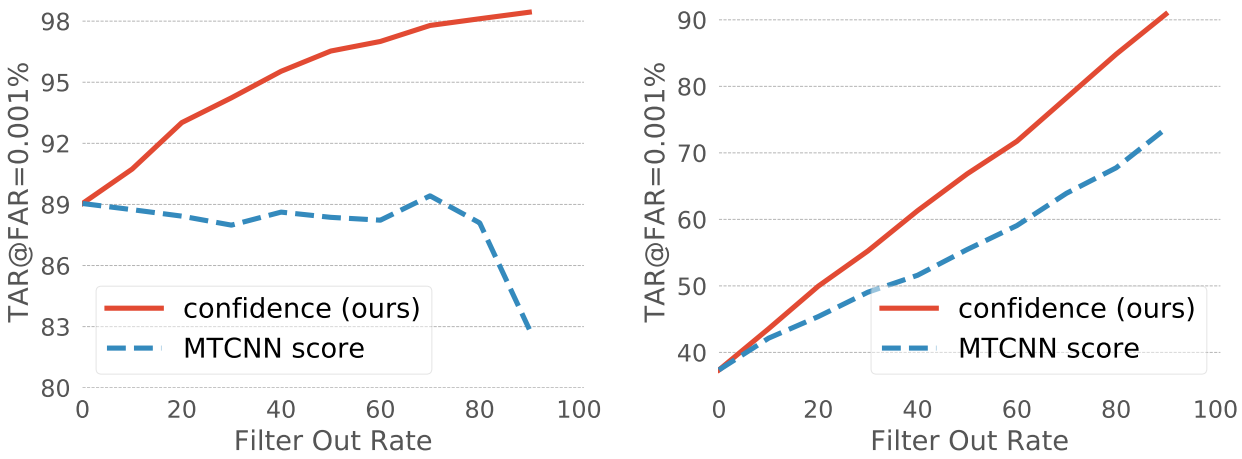

Figure 3.8 Visualization results on a high-quality, a low-quality and a mis-detected image from IJB-A. For each input, 5 images are reconstructed by a pre-trained decoder using the mean and 4 randomly sampled $\mathbf{z}$ vectors from the estimated distribution $p(\mathbf{z}|\mathbf{x})$.

common solution for this is to filter the images with a quality assessment tool. We show that PFE provides a natural solution for this task. We take all the images from LFW and IJB-A datasets for image-level face verification (We do not follow the original protocols here). The system is allowed to "filter out" a proportion of all images to maintain a better performance. We then report the TAR@FAR= 0.001% against the "Filter Out Rate". We consider two criteria for filtering: (1) the detection score of MTCNN [7] and (2) a confidence value predicted by our uncertainty module. Here the confidence for $i^{\text{th}}$ sample is defined as the inverse of harmonic mean of $\boldsymbol{\sigma}_i$ across all dimensions. For fairness, both methods use the original deterministic embedding representations and cosine similarity for matching. To avoid saturated results, we use the model trained on CASIA-WebFace with AM-Softmax. The results are shown in Figure 3.10. As one can see, the predicted confidence value is a better indicator of the potential recognition accuracy of the input image. This is an expected result since PFE is trained under supervision for the particular model while an external quality estimator is unaware of the kind of features used for matching by the model. Example images with high/low confidence/quality scores are shown in Figure 3.9.




(a) Ours (b) MTCNN

Figure 3.9 Example images from LFW and IJB-A that are estimated with the highest (H) confidence/quality scores and the lowest (L) scores by our method and MTCNN face detector.




(a) LFW (b) IJB-A

Figure 3.10 Comparison of verification performance on LFW and IJB-A (not the original protocol) by filtering a proportion of images using different quality criteria.

## 3.9 Conclusion

We have proposed probabilistic face embeddings (PFEs), which represent face images as distributions in the latent space. Probabilistic solutions were derived to compare and aggregate the PFE of face images. Unlike deterministic embeddings, PFEs do not suffer from the feature ambiguity dilemma for unconstrained face recognition. Quantitative and qualitative analysis on different settings showed that PFEs can effectively improve the face recognition performance by converting deterministic embeddings to PFEs. We have also shown that the uncertainty in PFEs is a good indicator for the "discriminative"quality of face images. In the future work we will explore how to learn PFEs in an end-to-end manner and how to address the data dependency within face templates.

# Chapter 4

# Universal Face Representation Learning

In this chapter, we will talk about the challenges faced by the feature extraction module in AFR systems and a potential solution to solve it. Almost all modern AFR systems use deep convolutional neural networks as the feature extraction module. As a black box function, such deep neural networks are trained to map input images to a feature space with small intra-identity distance and large inter-identity distance, which has been achieved by prior works through loss design and datasets with rich within-class variations [30, 32, 34, 36, 38]. However, even very large public datasets manifest strong biases, such as ethnicity [96, 97] or head poses [98, 99, 100]. This lack of variation leads to significant performance drops on challenging test datasets, for example, accuracy reported by prior state-of-the-art [54] on IJB-S or TinyFace [1, 6] are about 30% lower than IJB-A [5] or LFW [3].

Recent works seek to close the domain gap caused by such data bias through domain adaptation, i.e., identifying specific factors of variation and augmenting the training datasets [99], or further leveraging unlabeled data along such nameable factors [96]. While nameable variations are hard to identify exhaustively, prior works have sought to align the feature space between source and target domains [101, 97]. Alternatively, individual models might be trained on various datasets and ensembles to obtain good performance on each [102]. All these approaches either only handle specific variations, or require access to test data distributions, or accrue additional run-time complexity to handle wider variations. In contrast, we propose learning a single "universal" deep

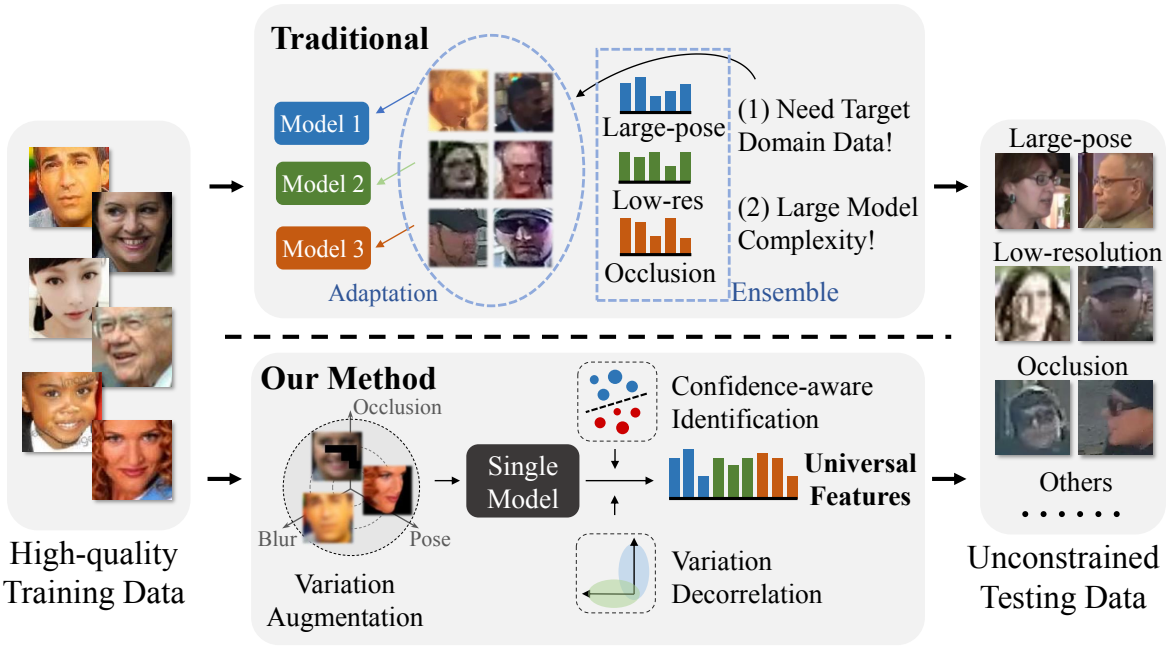



Figure 4.1 Traditional recognition models require target domain data to adapt from the high-quality training data to conduct unconstrained/low-quality face recognition. Model ensemble is further needed for a universal representation purpose which significantly increases model complexity. In contrast, our method works only on original training data without any target domain data information, and can deal with unconstrained testing scenarios.

feature representation that handles the variations in face recognition without requiring access to test data distribution and retains run-time efficiency, while achieving strong performance across diverse situations especially on low-quality images (see Figure 4.1).

This chapter introduces several novel contributions in Section 4.2 to learn such a universal representation. First, we note that inputs with non-frontal poses, low resolutions and heavy occlusions are key nameable factors that present challenges for "in-the-wild" applications, for which training data may be synthetically augmented. But directly adding hard augmented examples into training leads to a more difficult optimization problem. We mitigate this by proposing an identification loss that accounts for per-sample confidence to learn a probabilistic feature embedding. Second, we seek to maximize representation power of the embedding by decomposing it into sub-embeddings, each of which has an independent confidence value during training. Third, all the sub-embeddings are encouraged to be further decorrelated through two complementary regularization over different partitions of the sub-embeddings, i.e., classification loss on variations and adversarial loss on

different partitions. Fourth, we achieve further decorrelation by mining for additional variations for which synthetic augmentation is non-trivial. Finally, we account for the varying discrimination power of sub-embeddings for various factors through a probabilistic aggregation that accounts for their uncertainties.

In Section 4.4, we extensively evaluate the proposed methods on public datasets. Compared to our baseline model, the proposed method maintains the high accuracy on general face recognition benchmarks, such as LFW and YTF, while significantly boosting the performance on challenging datasets such as IJB-C, IJB-S, where new state-of-the-art performance is achieved. Detailed ablation studies show the impact of each of the above contributions in achieving these strong performance.

In summary, the main contributions of this chapter are:

- A method for learning a universal face representation by associating features with different variations, leading to improved generalization on diverse testing datasets.
- A confidence-aware identification loss that utilizes sample confidence during training to leverage hard samples.
- A feature decorrelation regularization that applies both a classification loss on variations and an adversarial loss on different partitions of the feature sub-embeddings, leading to improved performance.
- A training strategy to effectively combine synthesized data to train a face representation applicable to images outside the original training distribution.
- State-of-the-art results on several challenging benchmarks, such as IJB-A, IJB-C, TinyFace and IJB-S.

## 4.1 Related Work

Universal representation refers to a single model that can be applied to various visual domains (usually different tasks), e.g. object, character, road signs, while maintaining the performance of using a set of domain-specific models [103, 104, 105, 106, 97]. The features learned by such a

(a) Blur (b) Occlusion

(c) Pose (d) Randomly Combined

Figure 4.2 Samples from MS-Celeb-1M [2]with augmentation alongside different variations.

single model are believed to be more universal than domain-specific models. Different from domain generalization [107, 108, 109, 110, 111], which targets adaptability on unseen domains by learning from various seen domains, universal representation learning does not involve re-training on unseen domains. Several methods focus on increasing the parameter efficiency by reducing the domain-shift with techniques such as conditioned BatchNorm [103] and residual adapters [104, 105]. Based on SE modules [112], [106] propose a domain-attentive module for intermediate (hidden) features of a universal object detection network. Our work is different from those methods in two ways: (1) it is a method for similarity metric learning rather than detection or classification tasks and (2) it is model-agnostic. The features learned by our model can then be directly applied to different domains by computing the pairwise similarity between samples of unseen classes.

## 4.2 Proposed Approach

In this section, we first introduce three augmentable variations, namely blur, occlusion and head pose, to augment the training data. Visual examples of augmented data are shown in Figure 4.2 and the details can be found in Section 4.3. Then in Section 4.2.1, we introduce a confidence-aware identification loss to learn from hard examples, which is further extended in Section 4.2.2 by splitting the feature vectors into sub-embeddings with independent confidence. In Section 4.2.3, we apply

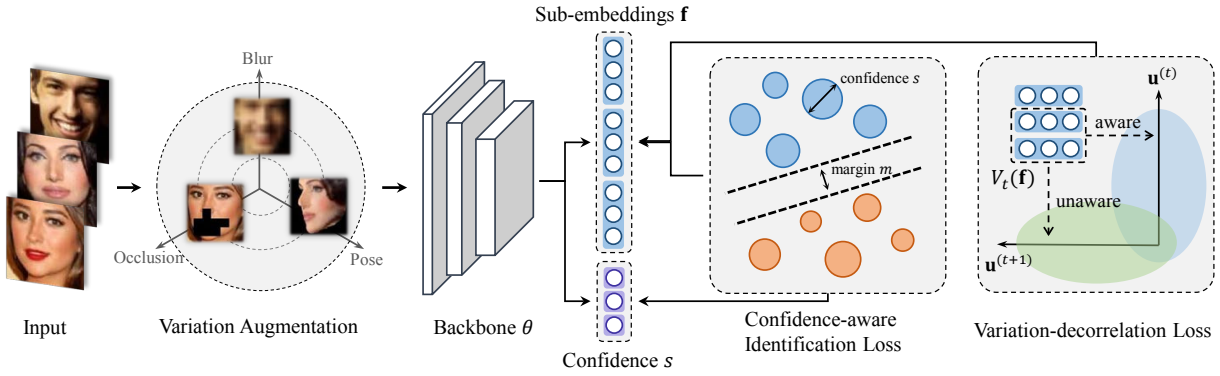



Figure 4.3 Overview of the proposed method. High-quality input images are first augmented according to pre-defined variations, i.e., blur, occlusion and pose. The feature representation is then split into sub-embeddings associated with sample-specific confidences. Confidence-aware identification loss and variation decorrelation loss are developed to learn the sub-embeddings.

the introduced augmentable variations to further decorrelate the feature embeddings. A method for discovering further non-augmentable variations is proposed to achieve better decorrelation. Finally, an uncertainty-guided pairwise metric is proposed for inference.

### 4.2.1 Confidence-Aware Identification Loss

We investigate the posterior probability of being classified to identity $j \in \{1, 2, \ldots, N\}$, given the input sample $\mathbf{x}_i$. Denote the feature embedding of sample $i$ as $\mathbf{f}_i$ and the $j^{\text{th}}$ identity prototype vector as $\mathbf{w}_j$, which is the identity template feature. A probabilistic embedding network $\theta$ represents each sample $\mathbf{x}_i$ as a Gaussian distribution $\mathcal{N}(\mathbf{f}_i, \sigma_i^2 \mathbf{I})$ in the feature space. The likelihood of $\mathbf{x}_i$ being a sample of class $j$ is given by:

$$\begin{aligned} p(\mathbf{x}_i | y = j) &\propto p_\theta(\mathbf{w}_j | \mathbf{x}_i) \\ &= \frac{1}{(2\pi\sigma_i^2)^{\frac{D}{2}}} \exp\left(-\frac{\left\|\mathbf{f}_i - \mathbf{w}_j\right\|^2}{2\sigma_i^2}\right), \end{aligned} \tag{4.1}$$

where $D$ is feature dimension. Further assuming the prior of assigning a sample to any identity as equal, the posterior of $\mathbf{x}_i$ belonging to the $j^{\text{th}}$ class is derived as:

$$\begin{aligned} p(y=j|\mathbf{x}_i) =& \frac{p(\mathbf{x}_i|y=j)p(y=j)}{\sum_{c=1}^{N} p(\mathbf{x}_i|y=c)p(y=c)} \\ =& \frac{\exp(-\frac{\left\|\mathbf{f}_i-\mathbf{w}_j\right\|^2}{2\sigma_i^2})}{\sum_{c=1}^{N}\exp(-\frac{\|\mathbf{f}_i-\mathbf{w}_c\|^2}{2\sigma_i^2})}, \end{aligned} \tag{4.2}$$

For simplicity, let us define a confidence value $s_i = \frac{1}{\sigma_i^2}$. Constraining both $\mathbf{f}_i$ and $\mathbf{w}_j$ on the $\ell_2$-normalized unit sphere, we have $\frac{\left\|\mathbf{f}_i-\mathbf{w}_j\right\|^2}{2\sigma_i^2} = s_i(1-\mathbf{w}_j^T\mathbf{f}_i)$ and

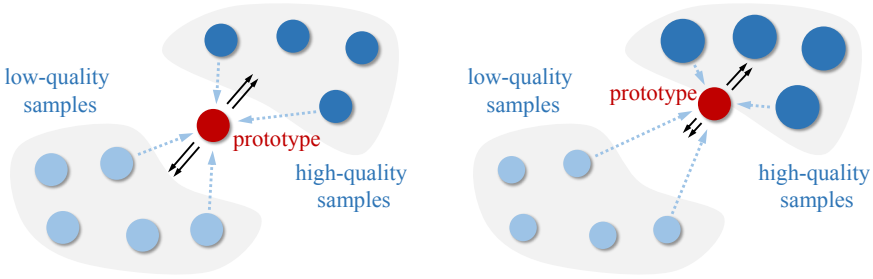
$$p_(y=j|\mathbf{x}_i) = \frac{\exp(s_i\mathbf{w}_j^T\mathbf{f}_i)}{\sum_{c=1}^{N}\exp(s_i\mathbf{w}_c^T\mathbf{f}_i)}. \tag{4.3}$$

The effect of confidence-aware posterior in Equation 4.3 is illustrated in Figure 4.4. When training is conducted among samples of various qualities, if we assume the same confidence across all samples, the learned prototype will be in the center of all samples. This is not ideal, as low-quality samples convey more ambiguous identity information. In contrast, if we set up sample-specific confidence $s_i$, where high-quality samples show higher confidence, it will push the prototype $\mathbf{w}_j$ to be more similar to high-quality samples in order to maximize the posterior. Meanwhile, during update of the embedding $\mathbf{f}_i$, it provides a stronger push for low-quality $\mathbf{f}_i$ to be closer to the prototype.

Adding loss margin [36] over the exponential logit has been shown to be effective in narrowing the within-identity distribution. We also incorporate it into our loss:

$$\mathcal{L}'_{idt} = -\log\frac{\exp(s_i\mathbf{w}_{y_i}^T\mathbf{f}_i - m)}{\exp(s_i\mathbf{w}_{y_i}^T\mathbf{f}_i - m) + \sum_{j\neq y_i}\exp(s_i\mathbf{w}_j^T\mathbf{f}_i)}, \tag{4.4}$$

where $y_i$ is the ground-truth label of $\mathbf{x}_i$. Our confidence-aware identification loss (C-Softmax) is different from cosine loss[36] as follows: (1) each image has an independent and dynamic $s_i$ rather than a constant shared scalar and (2) the margin parameter $m$ is not multiplied by $s_i$. The




(a) w/o confidence (b) w/ confidence

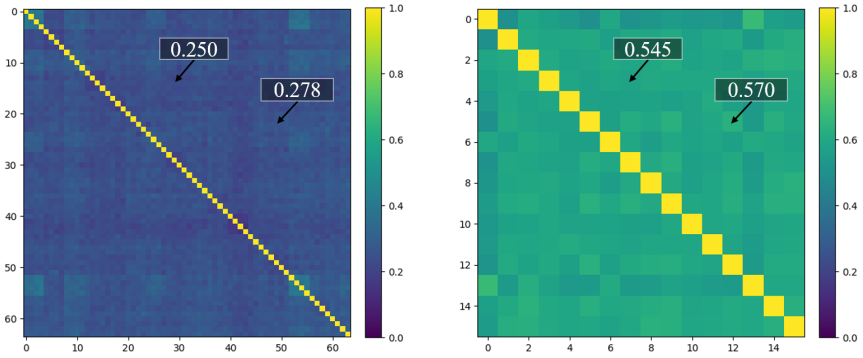
Figure 4.4 Illustration of confidence-aware embedding learning on quality-various data. With confidence guiding, the learned prototype is closer to high-quality samples which represents the identity better.

independence of $s_i$ allows it to gate the gradient signals of $\mathbf{w}_j$ and $\mathbf{f}_i$ during network training in a sample-specific way, as the confidence (degree of variation) of training samples can have large differences. Though samples are specific, we aim to learn a homogeneous feature space such that the metric across different identities is consistent. Thus, allowing $s_i$ to compensate for the confidence difference of the samples, we expect $m$ to be consistently shared across all the identities.

### 4.2.2 Confidence-Aware Sub-Embeddings

Though the embedding $\mathbf{f}_i$ learned through a sample-specific gating $s_i$ can deal with sample-level variations, we argue that the correlation among the entries of $\mathbf{f}_i$ itself is still high. To maximize the representation power and achieve a compact feature size, decorrelating the entries of the embedding is necessary. This encourages us to further break the entire embedding $\mathbf{f}_i$ into partitioned sub-embeddings, each of which is further assigned a scalar confidence value.

Illustrated in Figure 4.3, we partition the entire feature embedding $\mathbf{f}_i$ into $K$ equal-length sub-embeddings as in Equation 4.5. Accordingly, the prototype vector $\mathbf{w}_j$ and the confidence scalar




(a) sub-embedding of size 8 (b) sub-embedding of size 32

Figure 4.5 The correlation matrices of sub-embeddings by splitting the feature vector into different sizes. The correlation is computed in terms of distance to class center.

$s_i$ are also partitioned into the same size $K$ groups.

$$\begin{aligned} \mathbf{w}_j &= [\mathbf{w}_j^{(1)T}, \mathbf{w}_j^{(2)T}, \ldots, \mathbf{w}_j^{(K)T}], \\ \mathbf{f}_i &= [\mathbf{f}_i^{(1)T}, \mathbf{f}_i^{(2)T}, \ldots, \mathbf{f}_i^{(K)T}], \\ \mathbf{s}_i &= [s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(K)}], \end{aligned} \tag{4.5}$$

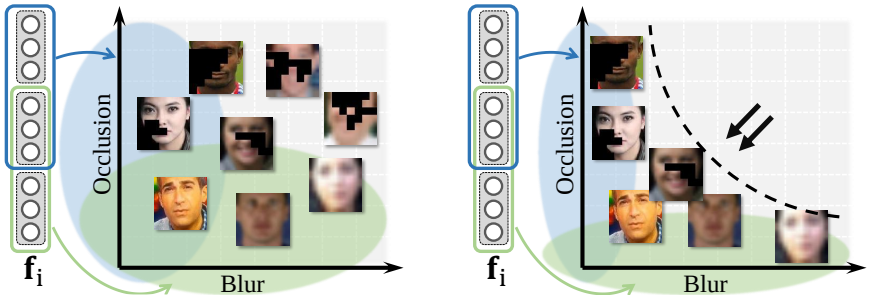
Each group of sub-embeddings $\mathbf{f}_i^{(k)}$ is $\ell_2$ normalized onto unit sphere separately. The final identification loss thus is:

$$\mathcal{L}_{idt} = -\log \frac{\exp(\mathbf{a}_{i,y_i} - m)}{\exp(\mathbf{a}_{i,y_i} - m) + \sum_{j \neq y_i} \exp(\mathbf{a}_{i,j})}, \tag{4.6}$$

$$\mathbf{a}_{i,j} = \frac{1}{K} \sum_{k=1}^{K} s_i^{(k)} \mathbf{w}_j^{(k)T} \mathbf{f}_i^{(k)}. \tag{4.7}$$

A common issue for neural networks is that they tend to be "over-confident" on predictions [113]. We add an additional $l_2$ regularization to constrain the confidence from growing arbitrarily large:

$$\mathcal{L}_{reg} = \frac{1}{K} \sum_{k=1}^{K} s_i^{(k)2}. \tag{4.8}$$




(a) variation-correlated features (b) variation-decorrelated features

Figure 4.6 The variation decorrelation loss disentangles different sub-embeddings by associating them with different variations. In this example, the first two sub-embeddings are forced to be invariant to occlusion while the second two sub-embeddings are forced to be invariant to blur. By pushing stronger invariance for each variation, the correlation/overlap between two variations is reduced.

### 4.2.3 Sub-Embeddings Decorrelation

Setting up multiple sub-embeddings alone does not guarantee the features in different groups are learning complementary information. Empirically shown in Figure 4.5, we find the sub-embeddings are still highly correlated, i.e., dividing $\mathbf{f}_i$ into equal 16 groups, the average correlation among all the sub-embeddings is 0.57. If we penalize the sub-embeddings with different regularization, the correlation among them can be reduced. By associating different sub-embeddings with different variations, we conduct variation classification loss on a subset of all the sub-embeddings while conducting variation adversarial loss in terms of other variation types. Given multiple variations, such two regularization terms are forced on different subsets, leading to better sub-embedding decorrelation.

For each augmentable variation $t \in \{1, 2, \ldots, M\}$, we generate a binary mask $V_t$, which selects a random $\frac{K}{2}$ subset of all sub-embeddings while setting the other half to be zeros. The masks are generated at the beginning of the training and will remain fixed during training. We guarantee that for different variations, the masks are different. We expect $V_t(\mathbf{f}_i)$ to reflect $t^{\text{th}}$ variation while invariant to the others. Accordingly, we build a multi-label binary discriminator $C$ by learning to

predict all variations from each masked subset:

$$\begin{aligned}\min_{C} \mathcal{L}_C &= -\sum_{t=1}^{M} \log p_C(\mathbf{u}_i = \hat{\mathbf{u}}_i | V_t(\mathbf{f}_i)) \\ &= -\sum_{t=1}^{M}\sum_{t'=1}^{M} \log p_C(u_i^{(t')} = \hat{u}_i^{(t')} | V_t(\mathbf{f}_i))\end{aligned} \tag{4.9}$$

where $\mathbf{u}_i = [u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(M)}]$ are the binary labels (0/1) of the known variations and $\hat{\mathbf{u}}_i$ is the ground-truth label. For example, if $t = 1$ corresponds to resolution, $\hat{u}_i^{(1)}$ would be 1 and 0 for high/low-resolution images, respectively. Note that Equation 4.9 is only used for training the discriminator $C$. The corresponding classification and adversarial loss of the embedding network is then given by:

$$\mathcal{L}_{cls} = -\sum_{t=1}^{M} \log p_C(u^{(t)} = \hat{u}_i^{(t)} | V_t(\mathbf{f}_i)) \tag{4.10}$$

$$\begin{aligned}\mathcal{L}_{adv} = -\sum_{t=1}^{M}\sum_{t'\neq t} (&\frac{1}{2} \log p_C(u^{(t')} = 0 | V_t(\mathbf{f}_i)) + \\ &\frac{1}{2} \log p_C(u^{(t')} = 1 | V_t(\mathbf{f}_i)))\end{aligned} \tag{4.11}$$

The classification loss $\mathcal{L}_{cls}$ to encourage $V_t$ to be variation-specific while $\mathcal{L}_{adv}$ is an adversarial loss to encourage invariance to the other variations. As long as no two masks are the same, it guarantees that the selected subsets $V_t$ is functionally different from other $V_{t'}$. We thus achieve decorrelation between $V_t$ and $V_{t'}$. The overall loss function for each sample is:

$$\min_{\theta} \mathcal{L} = \mathcal{L}_{idt} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{adv}\mathcal{L}_{adv}. \tag{4.12}$$

During the optimization, Equation (4.12) is averaged across the samples in the mini-batch.

### 4.2.4 Mining for Further Variations

The limited number (three in our method) of augmentable variations leads to limited effect of decorrelation as the number of $V_t$ are too small. To further enhance the decorrelation, as well to introduce more variations for better generalization ability, we aim to explore more variations with semantic meaning. Notice that not all the variations are easy to conduct data augmentation, e.g. smiling or not is hard to augment. For such variations, we attempt to mine out the variation labels from the original training data. In particular, we leverage an off-the-shelf attribute dataset CelebA [114] to train a attribute classification model $\theta_A$ with identity adversarial loss:

$$\begin{aligned} &\min_{\theta_A} \mathcal{L}_{\theta_A} = -\log p(l_A|\mathbf{x}_A) - \frac{1}{N_A}\sum_{c}^{N_A} \log p(y_A = c|\mathbf{x}_A) \\ &\min_{D_A} \mathcal{L}_{D_A} = -\log p(y_A = y_{\mathbf{x}_A}|\mathbf{x}_A), \end{aligned} \tag{4.13}$$

where $l_A$ is the attribute label and $y_A$ is the identity label. $\mathbf{x}_A$ is the input face image and $N_A$ is the number of identities in the CelebA dataset. The first term penalizes the feature to classify facial attributes and the second term penalizes the feature to be invariant to identities.

The attribute classifier is then applied to the recognition training set to generate $T$ new soft variation labels, e.g. smiling or not, young or old. These additional variation binary labels are merged with the original augmentable variation labels as: $\mathbf{u}_i = [u_i^{(1)}, \ldots, u_i^{(M)}, u_i^{(M+1)}, \ldots, u_i^{(M+T)}]$ and are then incorporated into the decorrelation learning framework in Section 4.2.3.

### 4.2.5 Uncertainty-Guided Probabilistic Aggregation

Considering the metric for inference, simply taking the average of the learned sub-embeddings is sub-optimal. This is because different sub-embeddings show different discriminative power for different variations. Their importance should vary according to the given image pairs. Inspired by [54], we consider applying the uncertainty associated with each embedding for a pairwise

similarity score:

$$score(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{2}\sum_{k=1}^{K}\frac{\left\|\mathbf{f}_i^{(k)} - \mathbf{f}_j^{(k)}\right\|^2}{\sigma_i^{(k)2} + \sigma_j^{(k)2}} - \frac{D}{2K}\sum_{k=1}^{K}\log(\sigma_i^{(k)2} + \sigma_j^{(k)2}) \tag{4.14}$$

Though with Equation 4.8 for regularization, we empirically find that the confidence learned with the identification loss still tend to be overconfident and hence cannot be directly used for Equation 4.14, so we fine-tune the original confidence branch to predict $\sigma$ while fixing the other parts. We refer the readers to [54] for the training details of fine-tuning.

## 4.3 Implementation Details

**Training Details and Baseline** All the models are implemented with Pytorch v1.1. We use the clean list from ArcFace [38] for MS-Celeb-1M [2] as training data. After cleaning the overlapped subjects with the testing sets, we have 4.8M images of 76.5K classes. We use the method in [115] for face alignment and crop all images into a size of $110 \times 110$. Random and center cropping are applied during training and testing, respectively, to transform the images into $100 \times 100$. The backbone of our embedding network $\theta$ is a modified 100-layer ResNet in [38]. The network is split into two different branches after the last convolution layer, each of which includes one fully connected layer. The first branch outputs a 512-D vector, which is further split into 16 sub-embeddings. The other branch outputs a 16-D vector, which are confidence values for the sub-embeddings. The exp function is used to guarantee all the confidence values $s_i^{(k)}$ are positive. The model $\theta_A$ that we used for mining additional variations is a four layer CNN. The four layers have 64, 128, 256 and 512 kernels, respectively, all of which are $3 \times 3$. The embedding size is 512 for all models, and the features are split into 16 groups for multi-embedding methods. The model $C$ is a linear classifier. The baseline models in the experiments are trained with CosFace loss function [36, 35],

which achieves state-of-the-art performance on general face recognition tasks. The models without domain augmentation are trained for 18 epochs and models with domain augmentation are trained for 27 epochs to ensure convergence. We empirically set $\lambda_{reg}$, $\lambda_{cls}$ and $\lambda_{adv}$ as 0.001, 2.0 and 2.0, respectively. The margin $m$ is empirically set to 30. For non-augmentable variations, we choose $T = 3$ attributes, namely smiling, young and gender.

**Variation Augmentation** For the low-resolution, we use Gaussian blur with a kernel size between 3 and 11. For the occlusion, we split the images into $7 \times 7$ blocks and randomly replace some blocks with black masks. (3) For pose augmentation, we use PRNet [116] to fit the 3D model of near-frontal faces in the dataset and rotate them into a yaw degree between 40° and 60°. All the augmentations are randomly combined with a probability of 30% for each.

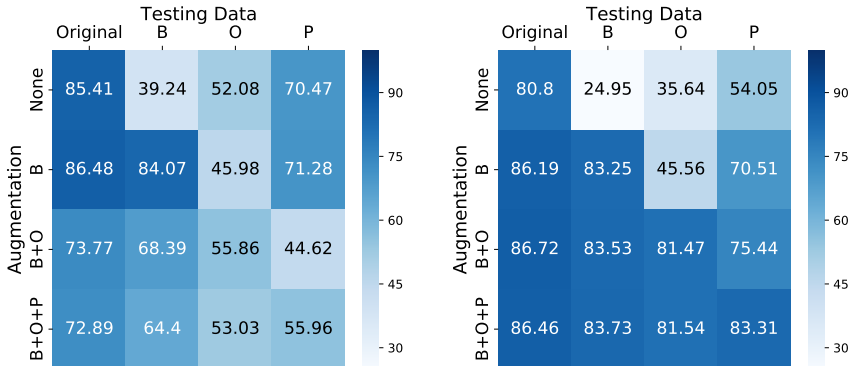
## 4.4 Experiments

In this section, we firstly introduce different types of datasets reflecting different levels of variation. Different levels of variation indicate different image quality and thus lead to different performance. Then we conduct detailed ablation study over the proposed confidence-aware loss and all the proposed modules. Further, we show evaluation on those different types of testing datasets and compare to state-of-the-art methods.

### 4.4.1 Datasets

We evaluate our models on eight face recognition benchmarks, covering different real-world testing scenarios. The datasets are roughly categorized into three types based on the level of variations:

**Type I: Limited Variation** LFW [3], CFP [4], YTF [51] and MegaFace [117] are four widely applied benchmarks for general face recognition. We believe the variations in those datasets are limited, as only one or few of the variations being presented. In particular, YTF are video samples with relatively lower resolution; CFP [4] are face images with large pose variation but of high



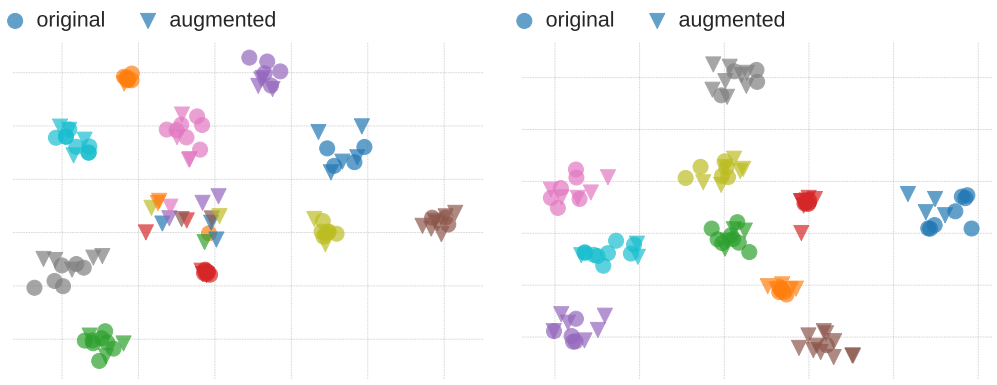

(a) Baseline (b) Proposed

Figure 4.7 Testing results on synthetic data of different variations from IJB-A benchmark (TAR@FAR=0.01%). Different rows correspond to different augmentation strategies during training. Columns are different synthetic testing data. "B", "O", "P" represents "Blur", "Occlusion" and "Pose", respectively. The performance of the proposed method is improved in a monotonous way with more augmentations being added.

resolution; MegaFace includes 1 million distractors crawled from internet while its labeled images are all high-quality frontal faces from FaceScrub dataset [118]. For both LFW and YTF, we use the unrestricted verification protocol. For CFP, we focus on the frontal-profile (FP) protocol. We test on both verification and identification protocols of MegaFace.

**Type II: Mixed Quality** IJB-A [5] and IJB-C [42] include both high quality celebrity photos taken from the wild and low quality video frames with large variations of illumination, occlusion, head pose, etc. We test on both verification and identification protocols of the two benchmarks.

**Type III: Low Quality** We test on TinyFace [6] and IJB-S [1], two extremely challenging benchmarks that are mainly composed of low-quality face images. In particular, TinyFace only consists of low-resolution face images captured in the wild, which also includes other variations such as occlusion and pose. IJB-S is a video face recognition dataset, where all images are video frames captured by surveillance cameras except a few high-quality registration photos for each person.




(a) Baseline (b) Proposed

Figure 4.8 t-SNE visualization of the features in a 2D space. Colors indicate the identities. Original training samples and augmented training samples are shown in circle and triangle, respectively.

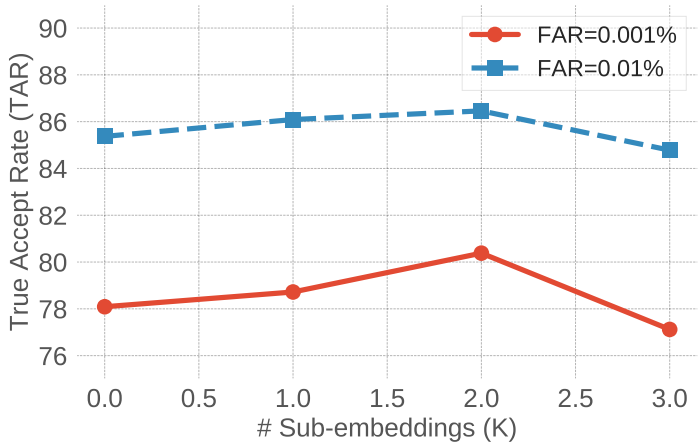



Figure 4.9 Performance change with respect to difference choice of K.

### 4.4.2 Ablation Study

#### Effect of Confidence-aware Learning

We train a set of models by gradually adding the nameable variations. The "Baseline" model is an 18-layer ResNet trained on a randomly selected subset of MS-Celeb-1M (0.6M images). The "Proposed" model is trained with the confidence-aware identification loss and $K = 16$ embedding groups. As a controlled experiment, we apply the same type of augmentation on IJB-A dataset to synthesize testing data of the corresponding variations. In Figure 4.7, "Baseline" model shows decreasing performance when gradually adding new variations as in the grid going down from

Table 4.1 Ablation study over the whole framework. VA: Variation Augmentation (Section 4.2), CI: Confidence-aware Identification loss (Section 4.2.1), ME: indicates Multiple Embeddings (Section 4.2.3), DE: Decorrelated Embeddings (Section 4.2.3), PA: Probabilistic Aggregation. (Section 4.2.5). E(all) uses all the proposed modules.

| Model | Method | | | | | LFW | CFP-FP | IJB-A (TAR@FAR) | | TinyFace | | IJB-S | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VA | CI | ME | DE | PA | Accuracy | Accuracy | FAR=0.001% | FAR=0.01% | Rank1 | Rank5 | Rank1 | Rank 5 |
| Baseline | | | | | | 99.75 | 98.16 | 82.20 | 93.05 | 46.75 | 51.79 | 37.14 | 46.75 |
| A | ✓ | | | | | 99.70 | 98.35 | 82.42 | 93.86 | 55.26 | 59.04 | 51.27 | 58.94 |
| B | ✓ | ✓ | | | | 99.78 | 98.30 | 94.70 | 96.02 | 57.11 | 63.09 | 59.87 | 66.90 |
| C | ✓ | ✓ | ✓ | | | 99.77 | 98.50 | 94.75 | 96.27 | 57.30 | 63.73 | 59.66 | 66.30 |
| | ✓ | ✓ | ✓ | | ✓ | 99.78 | **98.66** | **96.10** | 97.29 | 55.04 | 60.97 | 59.71 | 66.32 |
| D | | ✓ | ✓ | ✓ | | 99.65 | 97.77 | 80.06 | 92.14 | 34.76 | 39.86 | 29.87 | 40.69 |
| | | ✓ | ✓ | ✓ | ✓ | 99.68 | 98.00 | 94.37 | 96.42 | 35.05 | 40.13 | 50.00 | 56.27 |
| E (all) | ✓ | ✓ | ✓ | ✓ | | 99.75 | 98.30 | 95.00 | 96.27 | 61.32 | 66.34 | 60.74 | 66.59 |
| | ✓ | ✓ | ✓ | ✓ | ✓ | **99.78** | 98.64 | 96.00 | **97.33** | **63.89** | **68.67** | **61.98** | **67.12** |

Table 4.2 Our method compared to state-of-the-art methods on Type I datasets. The MegaFace verification rates are computed at FAR=0.0001%. "-" indicates that the author did not report the performance on the corresponding protocol.

| Method | LFW | YTF | CFP-FP | MF1 | |
|---|---|---|---|---|---|
| | | | | Rank1 | Veri. |
| FaceNet [30] | 99.63 | 95.1 | - | - | - |
| CenterFace [32] | 99.28 | 94.9 | - | 65.23 | 76.52 |
| SphereFace [34] | 99.42 | 95.0 | - | 75.77 | 89.14 |
| ArcFace [38] | 99.83 | 98.02 | 98.37 | 81.03 | 96.98 |
| CosFace [36] | 99.73 | 97.6 | - | 77.11 | 89.88 |
| Ours (Baseline) | 99.75 | 97.16 | 98.16 | 80.03 | 95.54 |
| Ours (Baseline+VA) | 99.70 | 97.10 | 98.36 | 78.10 | 94.31 |
| Ours (all) | 99.75 | 97.68 | 98.30 | 79.10 | 94.92 |
| Ours (all) + PA | 99.78 | 97.92 | 98.64 | 78.60 | 95.04 |

top row to bottom row. In comparison, the proposed method shows improving performance when adding new variations from top to bottom, which highlights the effect of our confidence-aware representation learning and it further allows to add more variations into the framework training.

We also visualize the features with t-SNE projected onto 2D embedding space. Figure 4.8 shows that for "Baseline" model, with different variation augmentations, the features actually are mixed and thus are erroneous for recognition. While for "Proposed" model, different variation augmentation generated samples are still clustered together to its original samples, which indicates that identity is well preserved. Under the same settings as above, we also show the effect of using different number of groups in Figure 4.9. At the beginning, splitting the embedding space into more

Table 4.3 Our model compared to state-of-the-art methods on IJB-A, IJB-C and IJB-S. "-" indicates that the author did not report the performance on the corresponding protocol. "*" indicates fine-tuning on the target dataset during evaluation on IJB-A benchmark and "+" indicates the testing performance by using the released models from corresponding authors.

| Method | IJB-A (Vrf) | | IJB-A (Idt) | IJB-C (Vrf) | | IJB-C (Idt) | IJB-S (S2B) | | |
|---|---|---|---|---|---|---|---|---|---|
| | FAR=0.001% | FAR=0.01% | Rank1 | FAR=0.001% | FAR=0.01% | Rank1 | Rank1 | Rank5 | FPIR=1% |
| NAN [88] | - | 88.1±1.1 | 95.8±0.5 | - | - | - | - | - | - |
| L2-Face [37] | 90.9±0.7 | 94.3±0.5 | 97.3±0.5 | - | - | - | - | - | - |
| DA-GAN [119] | 94.6±0.1 | 97.3±0.5 | **99.0±0.2** | - | - | - | - | - | - |
| [44] | - | 92.1±1.4 | 98.2±0.4 | 76.8 | 86.2 | 91.4 | - | - | - |
| Multicolumn [90] | - | 92.0±1.3 | - | 77.1 | 86.2 | - | - | - | - |
| ArcFace [38] | 93.7±1.0 | 94.2±0.8 | 97.0±0.6 | 93.5 | 95.8 | 95.87 | 57.36 | 64.95 | 41.23 |
| Ours (Baseline) | 82.6±8.3 | 93.3±3.0 | 95.5±0.7 | 43.9 | 86.7 | 89.85 | 37.14 | 46.75 | 24.75 |
| Ours (Baseline + VA) | 82.4±8.1 | 93.9±3.5 | 95.8±0.6 | 47.6 | 90.6 | 90.16 | 51.27 | 58.94 | 31.19 |
| Ours (all) | 95.0±0.9 | 96.3±0.6 | 97.5±0.4 | 91.6 | 93.7 | 94.39 | 60.74 | 66.59 | 37.11 |
| Ours (all) + PA | **96.0±0.8** | **97.3±0.4** | 97.5±0.3 | **95.0** | **96.6** | **96.00** | **61.98** | **67.12** | **42.73** |

groups increases performance for both TARs. When the size of each sub-embedding becomes too small, the performance starts to drop because of the limited capacity for each sub-embedding.

### Ablation on All Modules

We investigate each module's effect by looking into the ablative models in Table 4.1. Starting from the baseline, model A is trained with variation augmentation. Based on model A, we add confidence-aware identification loss to obtain model B. Model C is further trained by setting up multiple sub-embeddings. In model E, we further added the decorrelation loss. We also compare with a Model D with all the modules except variation augmentation. Model C, D and E, which have multiple embeddings, are tested w/ and w/o probabilistic aggregation (PA). The methods are tested on two type I datasets (LFW and CFP-FP), one type-II dataset (IJB-A) and one type-III dataset (TinyFace).

Shown in Table 4.1, compared to baseline, adding variation augmentation improves performance on CFP-FP, TinyFace, and IJBA. These datasets present exactly the variations introduced by data augmentation, i.e., pose variation and low resolution. However, the performance on LFW fluctuates from baseline as LFW is mostly good quality images with few variations. In comparison, model B and C are able to reduce the negative impact of hard examples introduced by data augmentation and leads to consistent performance boost across all benchmarks. Meanwhile, we observe that splitting into multiple sub-embeddings alone does not improve (compare B to C first row) significantly, which

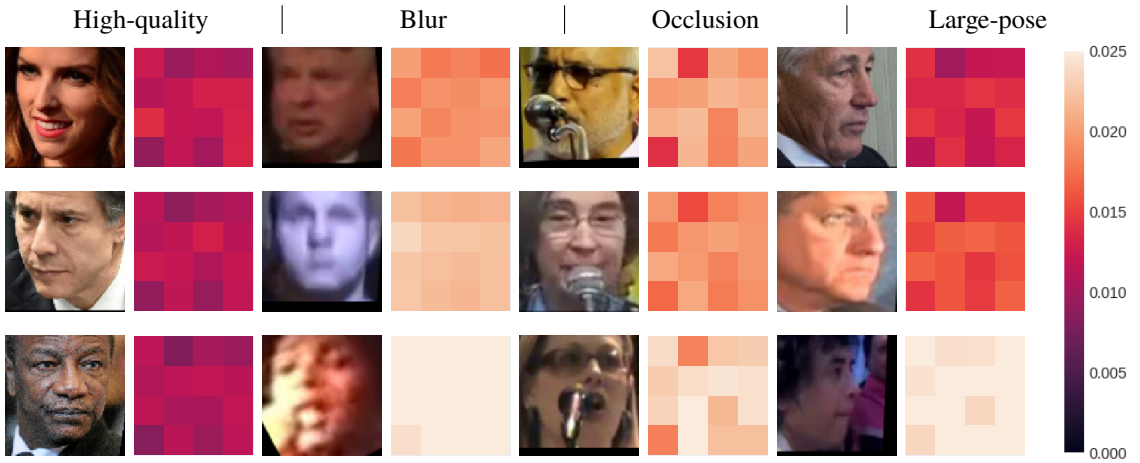



Figure 4.10 Heatmap visualization of sub-embedding uncertainty on different types of images from IJB-C dataset, shown on the right of each face image. 16 values are arranged in 4×4 grids (no spatial meaning). Brighter color indicates higher uncertainty.

can be explained by the strongly correlated confidence among the sub-embeddings (see Figure 4.5). Nevertheless, with the decorrelation loss and probabilistic aggregation, different sub-embeddings are able to learn and combine complementary features to further boost the performance, i.e., the performance in the second row of Model E is consistently better than its first row.

### 4.4.3 Evaluation on General Datasets

We compare our method with state-of-the-art methods on general face recognition datasets, i.e., those Type I datasets with limited variation and high quality. Since the testing images are mostly with good quality, there is limited advantage of our method which is designed to deal with larger variations. Even though, shown in Table 4.2, our method still stands on top being better than most of the methods while slightly worse than ArcFace. Notice that our baseline model already achieves good performance across all the testing sets. It actually verifies that the type I testing sets do not show significant domain gap from the training set, where even without variation augmentation or embedding decorrelation, the straight training can lead to good performance.

### 4.4.4 Evaluation on Mixed/Low Quality Datasets

When evaluating on more challenging datasets, those state-of-the-art general methods encounter performance drop as the challenging datasets present large variations and thus large domain gap from the good quality training datasets. Table 4.3 shows the performance on three challenging benchmarks: IJB-A, IJB-C and IJB-S. The proposed model achieves consistently better results than the state-of-the-arts. In particular, simply adding variation augmentation ("Ours (Baseline + VA)") actually leads to a worse performance on IJB-A and IJB-C. When variation augmentation is combined with our proposed modules ("Ours"), significant performance boost is achieved. Further adding PA with "Ours", we achieve even better performance across all datasets and protocols. Notice that IJB-A is a cross-validation protocol. Many works fine-tune on training splits before evaluation (shown with "*"). Even though, our method without fine-tuning still outperforms the state-of-the-art methods with significant margin on IJB-A verification protocol, which suggests that our method indeed learns the representation towards dealing with unseen variations.

Table 4.3 last column shows the evaluation on IJB-S, which is so far the most challenging benchmark targeting real surveillance scenario with severe poor quality images. We show the Surveillance-to-Booking (S2B) protocol of IJB-S. As IJB-S is recently released, there are few studies that have evaluated on this dataset. To comprehensively evaluate our model, we use the publicly released models from ArcFace [38] for comparison. Our method achieves consistently better performance across Rank-1 and Rank-5 identification protocol. For TinyFace, as in Table 4.1, we achieve 63.89%, 68.67% rank-1 and rank-5 accuracy, where [6] reports 44.80%, 60.40%, and ArcFace achieves 47.39%, 52.28%. Combining Table 4.2, our method achieves top level accuracy on general recognition datasets and significantly better accuracy on challenging datasets, which demonstrates the advantage in dealing with extreme or unseen variations.

**Uncertainty Visualization** Figure 4.10 shows uncertainty scores for the 16 sub-embeddings reshaped into $4 \times 4$ grids. High-quality and low-quality sub-embeddings are shown in dark and light colors respectively. The uncertainty map presents different patterns for different variations.

## 4.5 Conclusion

In this work, we propose a universal face representation learning framework, URFace, to recognize faces under all kinds of variations. We firstly introduce three nameable variations into MS-Celeb-1M training set via data augmentation. Traditional methods encounter convergence problem when directly feeding the augmented hard examples into training. We propose a confidence-aware representation learning by partitioning the embedding into multiple sub-embeddings and relaxing the confidence to be sample and sub-embedding specific. Further, the classification and adversarial losses on variations are proposed to decorrelate the sub-embeddings. By formulating the inference with an uncertainty model, the sub-embeddings are aggregated properly. Experimental results show that our method achieves top performance on general benchmarks such as LFW and MegaFace, and significantly better accuracy on challenging benchmarks such as IJB-A, IJB-C and IJB-S.

# Chapter 5

# Generalizing Face Representation with Unlabeled Images

## 5.1 Introduction

Machine learning algorithms typically assumes that training and testing data come from the same underlying distribution. However, in practice, we would often encounter testing domains that are different from the population where the training data is drawn. Since it is non-trivial to collect data for all possible testing domains, learning representations that are generalizable to heterogeneous testing data is desired [108, 120, 121, 122, 123]. Particularly for face recognition, this problem is reflected by the domain gap between the semi-constrained training datasets and unconstrained testing datasets. Nearly all of the state-of-the-art deep face networks are trained on large-scale web-crawled face images, most of which are high-quality celebrity photos [50, 2]. But in practice, we wish to deploy the trained FR systems for many other scenarios, e.g. unconstrained photos [5, 41, 42] and surveillance [1]. The large degree of face variation in the testing scenarios, compared to the training set, could result in significant performance drop of the trained face models [42, 1].

The simplest solution to such a domain gap problem is to collect a large number of unconstrained labeled face images from different sources. However, due to privacy issue and human-labeling cost, it

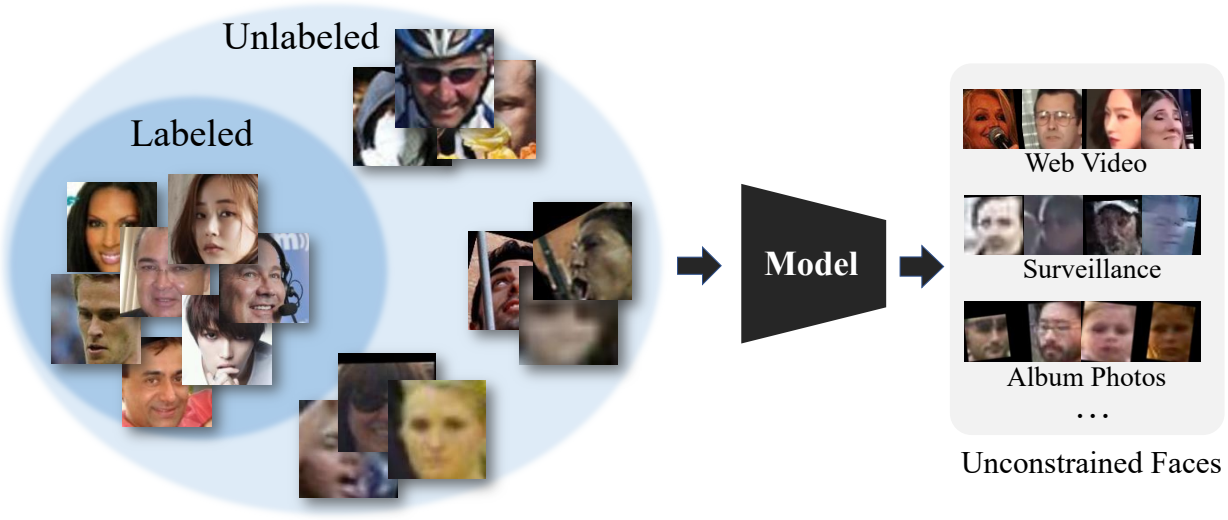



Figure 5.1 Illustration of the problem settings in our work. Blue circles imply the domains that the face images belong to. By utilizing diverse unlabeled images, we want to regularize the learning of the face embedding for more unconstrained face recognition scenarios.

is extremely hard to collect such a database. Other popular solutions to this problem include transfer learning and domain adaptation, which require domain-specific data to train a model for each of the target domains [124, 125, 126, 127, 128, 129]. However, in unconstrained face recognition, a face representation that is robust to all different kinds of variations is needed, so these domain-specific solutions are not appropriate. *Instead, it would be useful if we could utilize the commonly available, unlabeled data to achieve a domain-agnostic face representation that generalizes to unconstrained testing scenarios* (See Fig. 5.1). To achieve this goal, we would like to ask the following questions in this chapter:

- Is it possible to improve model generalizability to unconstrained faces by introducing more diversity from auxiliary unlabeled data?
- What kind of and how much unlabeled data do we need?
- How much performance boost could we achieve with the unlabeled data?

In this chapter, we propose such an semi-supervised framework for learning robust face representations. The unlabeled images are collected from a public face detection dataset, i.e. WiderFace [130], which contains more diverse types (sub-domains) of face images compared to

typical labeled face datasets used for training.

To utilize the unlabeled data, the proposed method jointly regularizes the embedding model from feature space and image space. We show that adversarial regularization can help to reduce domain gaps caused by facial variations, even in the absence of sub-domain labels. On the other hand, an image augmentation module is trained to discover the hidden sub-domain styles in the unlabeled data and apply them to the labeled training samples, thus increasing the discrimination power on difficult face examples. To our knowledge, this is the first study to use a heterogeneous unlabeled dataset to boost the model performance for general unconstrained face recognition. The contributions of this chapter are summarized as below:

- A semi-supervised learning framework for generalizing face representations with auxiliary unlabeled data.
- An multi-mode image translation module is proposed to perform data-driven augmentation and increase the diversity of the labeled training samples.
- Empirical results show that the regularization of unlabeled data helps to improve the recognition performance on challenging testing datasets, e.g. IJB-B, IJB-C, and IJB-S.

## 5.2 Related Work

### 5.2.1 Semi-supervised Learning

Classic semi-supervised learning involves a small number of labeled images and a large number of unlabeled images [131, 132, 133, 134, 135, 136, 137, 138]. The goal is to improve the recognition performance when we don't have sufficient data that are labeled. State-of-the-art semi-supervised learning methods can mainly be classified into four categories. (1) Pseudo-labeling methods generate labels for unlabeled data with the trained model and then use them for training [131]. In spite of its simplicity, it has been shown to be effective primarily for classification tasks where labeled data and unlabeled data share the same label space. (2) Temporal ensemble models maintain different

versions of model parameters to serve as teacher models for the current model [133, 134]. (3) Consistency-regularization methods apply certain types of augmentation to the unlabeled data while making sure the output prediction remains consistent after augmentation [132, 137, 138]. (4) Self-supervised learning, originally proposed for unsupervised learning, has recently been shown to be effective for semi-supervised learning as well [136]. Compared with classic semi-supervised learning addressed in the literature, our problem is different in two sense of heterogeneity: different domains and different identities between the labeled and unlabeled data. These differences make many classic semi-supervised learning methods unsuitable for our task.

### 5.2.2 Domain Adaptation and Generalization

In domain adaptation, the user has a dataset for a source domain and another for a fixed target domain [124, 125, 126, 128, 129]. If the target domain is unlabeled, this leads to an *unsupervised domain adaption* setting [125, 128, 129]. The goal is to improve the performance on the target domain so that it could match the performance on the source domain. This is achieved by reducing the domain gap between the two datasets in feature space. The problem about domain adaption is that one needs to acquire a new dataset and train a new model whenever there is a new target domain. In *domain generalization*, the user is given a set of labeled datasets from different domains. The model is jointly trained on these datasets so that it could better generalize to unseen domains [108, 120, 121, 122, 123]. Our problem shares the same goal with domain generalization methods: *we want to increase the model generalizability rather than improving performance on a specific target domain*. However, unlike domain generalization, we do not have identity labels for all the data, which makes our task even more difficult.

## 5.3 Methodology

Generally, in face representation learning, we are given a large labeled dataset $\mathcal{X}=\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i$ and $y_i$ are the face images and identity labels, re-

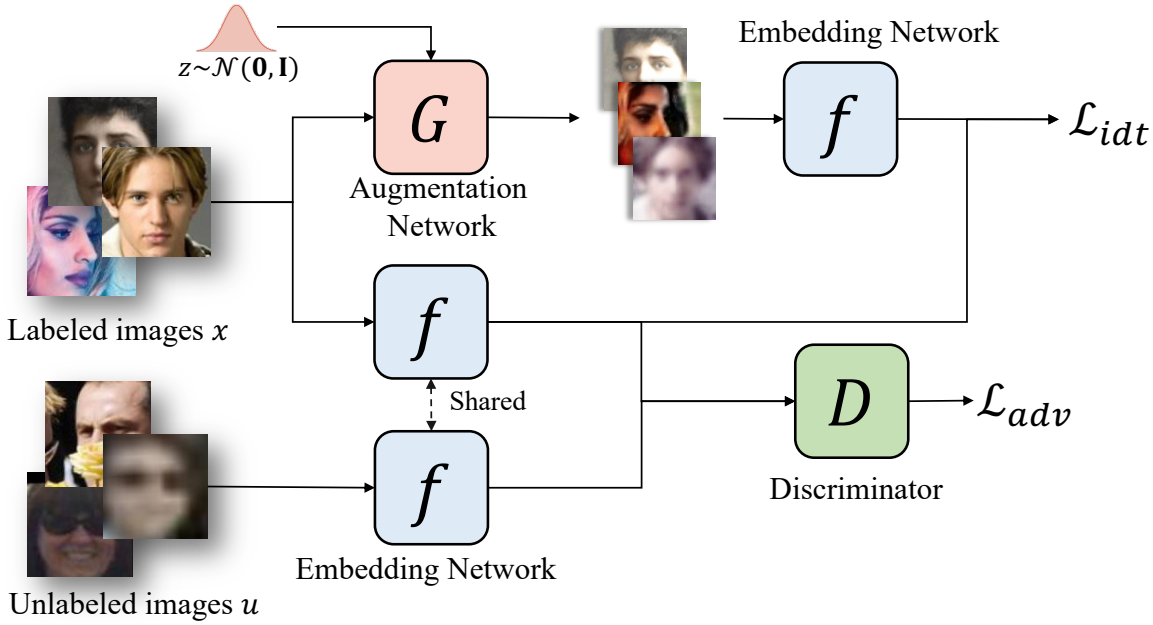
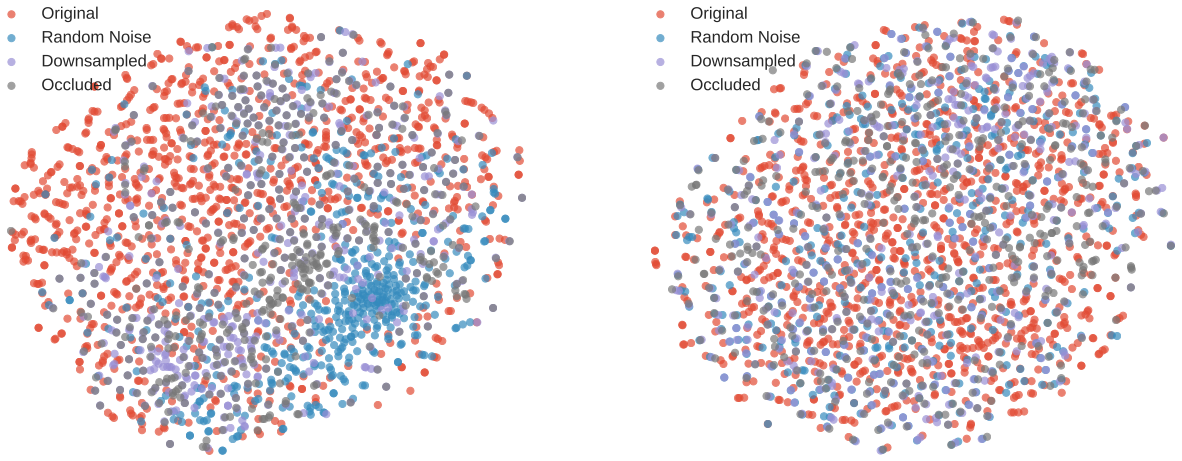



Figure 5.2 Overview of the training framework of the embedding network. In each mini-batch, a random subset of labeled data would be augmented by the augmentation network to introduce additional diversity. The non-augmented labeled data are used to train the feature discriminator. The adversarial loss forces the distribution of the unlabeled features to align with the labeled one.

spectively. The goal is to learn an embedding model $f$ such that $f(x)$ would be discriminative enough to distinguish between different identities. However, since $f$ is only trained on the domain defined by $\mathcal{X}$, which is usually semi-constrained celebrity photo, it might not generalize to unconstrained settings. In our framework, we assume the availability of another unlabeled dataset $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \ldots \mathcal{U}_k = \{u_1, u_2, \ldots, u_n\}$, collected from different sources (sub-domains). However, these sub-domain labels may not be available in real applications, thus we do not assume the access to them but instead seek solutions that could automatically leverage these hidden sub-domains. Then, we wish to simultaneously minimize three types of errors:

- Error due to discrimination power within the labeled domain $\mathcal{X}$.
- Error due to feature domain gap between the labeled domain $\mathcal{X}$ and the hidden sub-domains $\mathcal{U}_i$.
- Error due to discrimination power within the unlabeled domain $\mathcal{U}$.

An overview of the framework is shown in Fig. 5.2.




(a) w/o Domain Adversarial Loss

(b) w/ Domain Adversarial Loss

Figure 5.3 t-SNE visualization of the face embeddings using synthesized unlabeled images. Using part of the MS-Celeb-1M as unlabeled dataset, we create three sub domains by processing the images with either random Gaussian noise, random occlusion or downsampling. (a) different sub-domains show different domain shift in the embedding space of the supervised baseline. (b) with the holistic binary domain adversarial loss, each of the sub-domains is aligned with the distribution of the labeled data.

### 5.3.1 Minimizing Error in the Labeled Domain

The deep representation of a face image is usually a point in a hyper-spherical embedding space, where $\|f(x_i)\|^2 = 1$. State-of-the-art supervised face recognition methods all try to find an objective function to maximize the inter-class margin such that the representation could still be discriminative when tested on unseen identities. In this work, we choose to use CosFace loss function [36][35] for training the labeled images:

$$\mathcal{L}_{idt} = -\mathbb{E}_{x_i,y_i\sim\mathcal{X}}[\log \frac{e^{s(W_{y_i}^T f_i - m)}}{e^{s(W_{y_i}^T f_i - m)} + \sum_{j\neq y_i} e^{sW_{y_j}^T f_i}}]. \tag{5.1}$$

Here $s$ is the hyper-parameter controlling temperature, $m$ is a margin hyper-parameter and $W_j$ is the proxy vector of the $j^{th}$ identity in the embedding space, which is also $\ell_2$ normalized. We choose to use CosFace loss function because of its stability and high-performance. It could potentially be replaced by any other supervised identification loss function.

### 5.3.2 Minimizing Domain Gap

The unlabeled dataset $\mathcal{U}$ is assumed to be a diverse dataset collected from different sources, i.e. covering different sub-domains (types) of face images. If we have the access to such sub-domain labels, a natural solution to a domain-agnostic model would be aligning each of the sub-domains with the feature distribution of the labeled images. However, the sub-domain labels might not be available in many cases. In our experiment, we find there is no necessity for pairwise domain alignment. Instead, a binary domain alignment loss is sufficient to align the sub-domains. Formally, given a feature discriminator network $D$, we could reduce the domain gap via an adversarial loss:

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{x\sim\mathcal{X}}[\log D(y=0|f(x)] \\ &-\mathbb{E}_{u\sim\mathcal{U}}[\log D(y=1|f(u)],\end{aligned} \tag{5.2}$$

$$\begin{aligned}\mathcal{L}_{adv} = &-\mathbb{E}_{x\sim\mathcal{X}}[\log D(y=1|f(x)] \\ &-\mathbb{E}_{u\sim\mathcal{U}}[\log D(y=0|f(u)].\end{aligned} \tag{5.3}$$

The discriminator $D$ is a multi-layer binary classifier optimized by $\mathcal{L}_D$. It tries to learn a non-linear classification boundary between the two datasets while the embedding network needs to fool the discriminator by reducing the divergence between the distributions of $f(x)$ and $f(u)$. To see the effect of domain alignment loss, we conduct a controlled experiments with a toy dataset. We split the MS-Celeb-1M [2] dataset into labeled images and unlabeled images (no identity overlap). The unlabeled images are then processed with one of the three degradations: random Gaussian noise, random occlusion and downsampling. Thus, we create three sub-domains in the unlabeled dataset. The corresponding domain shift can be observed in the t-SNE plot in Fig. 5.3 (a), where the model is trained only on the labeled split. Then, we incorporate the augmented unlabeled images into training with the binary domain adversarial loss. In Fig. 5.3 (b), we observe that with the binary domain alignment loss, the distribution of each of sub-domains is aligned with the original domain, indicating reduced domain gaps.

### 5.3.3 Minimizing Error in the Unlabeled Domains

The domain alignment loss in Section 5.3.2 helps to eliminate the error caused by domain gaps between unconstrained faces. Thus, the remaining task is to improve the discrimination power of the face representation among the unlabeled faces. Many semi-supervised classification methods address this problem by using pseudo-labeling of unlabeled data [131, 137, 138], but this is not applicable to our problem since our unlabeled dataset does not share the same label space with the labeled one. Furthermore, because of data collection protocols, there is very little chance that one identity would have multiple unlabeled images. Thus, clustering-based methods are also infeasible for our task. Here, we consider to address this issue with a multi-mode augmentation method.

Prior studies have shown that an image translation network, such as CycleGAN [139], can be effectively used as a data augmentation module for domain adaptation [140]. The main idea of the augmentation network is to learn the difference between two domains in the image space and then augment the samples from source domain data to create training data with pseudo-labels in the target domain. Since our goal is to generalize the deep face representation to unconstrained faces, which involves a large variety, deterministic method such as CycleGAN would be unsuitable. Therefore, we propose to use a multi-mode image translation network that could discover the hidden domains in the unlabeled data and then augment the labeled training data with different styles. In particular, we need a function $G$ which maps labeled samples $x$ into the image space defined by the unlabeled faces, i.e. $p(x) \rightarrow p(u)$. Then, training the embedding $f$ on $G(x)$ could make it more discriminative in the image space defined by $U$. There are two requirement of the function $G$: (1) it should not change the identity of the input image and (2) it should be able to capture different styles that are present in the unlabeled images. Inspired by recent progress in image translation frameworks [139, 141], we propose to train $G$ as a style-transfer network that learns the visual styles during transfer in an unsupervised manner. The network $G$ can then be used as a data-driven augmentation module that generates diverse samples given an input from the labeled dataset. During the training, we randomly replace a subset of the labeled images to be augmented and put them into our identification learning framework. The details of training the augmentation network $G$ is given in Section 5.3.3.

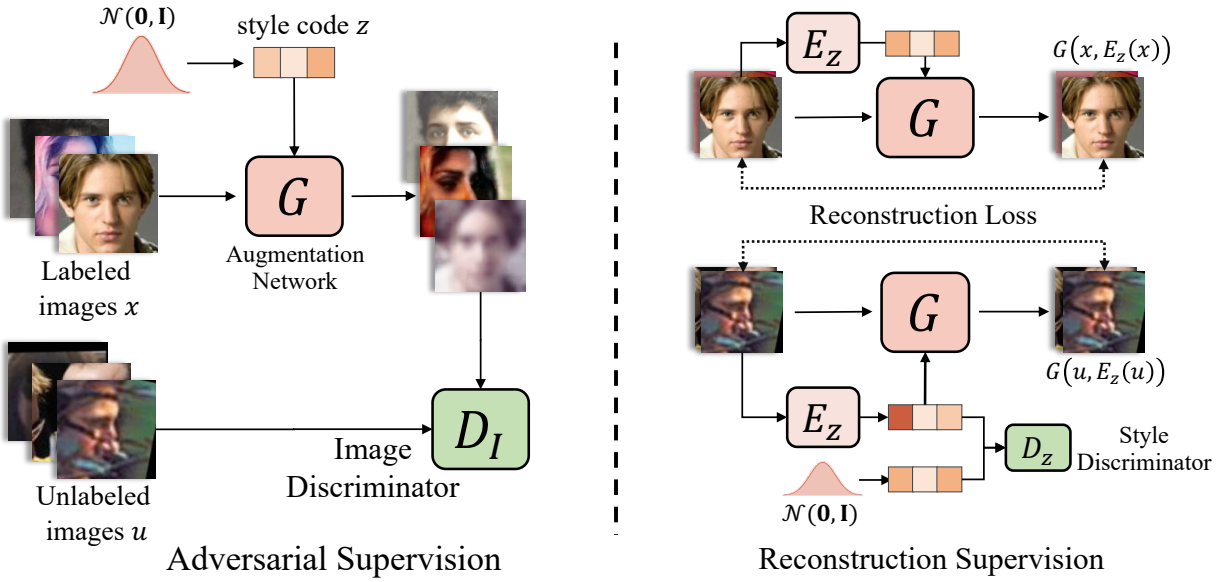



Figure 5.4 Training framework of the augmentation network $G$. The two pipelines are optimized jointly during training.

The overall loss function for the embedding network is given by:

$$\mathcal{L} = \lambda_{idt}\mathcal{L}_{idt} + \lambda_{adv}\mathcal{L}_{adv} \tag{5.4}$$

where $L_{idt}$ also includes the augmented labeled samples.

### Multi-mode Augmentation Network

The augmentation network $G$ is a fully convolutional network that maps an image to another. To preserve the geometric structure, our architecture does not involve any downsampling or upsampling. In order to generate styles similar to the unlabeled images, an image discriminator $D_I$ is trained to distinguish between the texture styles of unlabeled images and generated images:

$$\begin{aligned}\mathcal{L}_{D_I} = &- \mathbb{E}_{x\sim\mathcal{X}}[\log D_I(y=0|G(x,z))] \\ &- \mathbb{E}_{u\sim\mathcal{U}}[\log D_I(y=1|u)],\end{aligned} \tag{5.5}$$

$$\mathcal{L}^G_{adv} = - \mathbb{E}_{x\sim\mathcal{X}}[\log D_I(y=1|G(x,z))]. \tag{5.6}$$

Here $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random style vector to control the styles of the output image, which is injected into the generation process via Adaptive Instance Normalization (AdaIN) [142]. Although the adversarial learning could make sure the output are in the unlabeled space, but it cannot ensure that (1) the content of the input is maintained in the output image and (2) the random style $z$ is being used to generate diverse visual styles, corresponding to differnt sub-domains in the unlabeled images. We propose to utilize an additional reconstruction pipeline to simultaneously satisfy these two requirements. First, we introduce an additional style encoder $E_z$ to capture the corresponding style in the input image, as in [141]. A reconstruction loss is then enforced to keep the consistency of the image content:

$$\mathcal{L}_{rec}^{G} = \mathbb{E}_{x\sim\mathcal{X}}[\|x - G(x, E_z(x))\|^2] \tag{5.7}$$

$$+ \mathbb{E}_{u\sim\mathcal{U}}[\|u - G(u, E_z(u))\|^2], \tag{5.8}$$

Then, during the reconstruction, we add another latent style discriminator $D_z$ to guarantee the distribution of $E_z(u)$ align with prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\begin{aligned}\mathcal{L}_{D_z} = &- \mathbb{E}_{u\sim\mathcal{U}}[\log D_z(y = 0|E_z(u))] \\ &- \mathbb{E}_{z\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\log D_z(y = 1|z)],\end{aligned} \tag{5.9}$$

$$\mathcal{L}_{adv}^{z} = - \mathbb{E}_{u\sim\mathcal{U}}[\log D_z(y = 1|E_z(u))], \tag{5.10}$$

The overall loss function of the generator is given by:

$$\mathcal{L}^{G} = \lambda_{adv}^{G}\mathcal{L}_{adv}^{G} + \lambda_{rec}^{G}\mathcal{L}_{rec}^{G} + \lambda_{adv}^{z}\mathcal{L}_{adv}^{z} \tag{5.11}$$

A overview of the training framework of $G$ is given in Fig. 5.4 and example generated images are shown in Fig. 5.5.

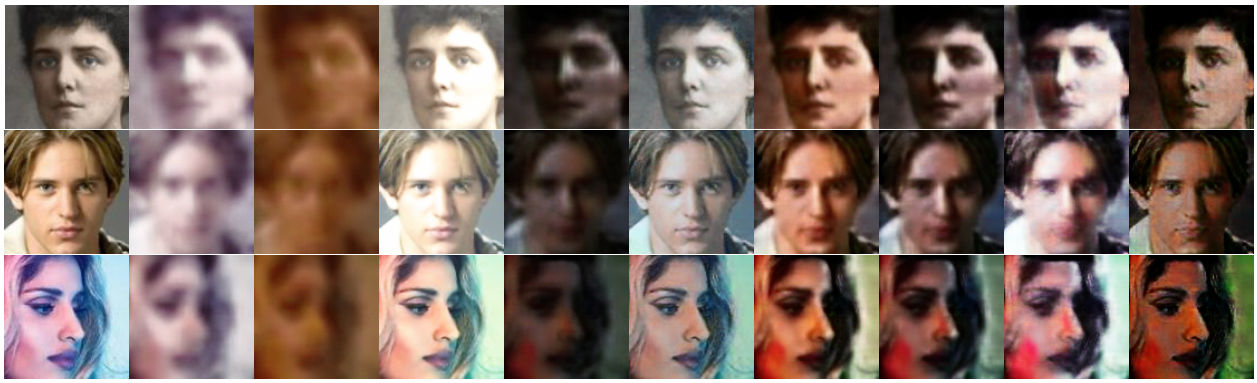


Figure 5.5 Example generated images of the augmentation network.

## 5.4 Experiments

### 5.4.1 Implementation Details

**Training Details of the Recognition Model** All the models are implemented with Pytorch v1.1. We use the RetinaFace [143] for face detection and alignment. All images are transformed into a size of $112 \times 112$. A modified 50-layer ResNet in [38] is used as our architecture. The embedding size is 512 for all models. By default, all the models are trained with $150,000$ steps with a batch size of 256. For semi-supervised models, we use 64 unlabeled images and 192 labeled images in each mini-batch. For models which uses the augmentation module, 20% of the labeled images are augmented by the generator network. The scale parameter $s$ and margin parameter $m$ are set to 30 and 0.5, respectively. We empirically set $\lambda_{idt}$, $\lambda_{adv}$ as 1.0 and 0.01. For models that utilizes the consistency regularization, $\lambda_{CR}$ is set to 0.2. Random image translation, flipping, occlusion and downsampling are used as data perturbation for those models.

**Training Details of the Generator Model** The generator is trained for $160,000$ steps with a batch size of 8 images (4 from each dataset). Adam optimizer is used with $\beta_1 = 0.5$ and $\beta_2 = 0.99$. The learning rate starts with $1e-4$ and drops to $1e-5$ after $80,000$ steps. $\lambda^G_{adv}$, $\lambda^G_{rec}$ and $\lambda^z_{adv}$ are set to as 1.0, 10.0 and 1.0, respectively. The architecture of the generator is based on MUNIT [141]. Let `c5s1-k` be a 5×5 convolutional layer with $k$ filters and stride 1. `dk-IN` denotes a 3×3 convolutional layer with $k$ filters and dilation 2, where IN means Instance Normalization [144]. Similarly, AdaIN means Adaptive Instance Normalization [142] and LN denotes Layer Normalization [145]. `fc8`

denotes a fully connected layer with 8 filters. `avgpool` denotes a global average pooling layer. No normalization is used in the style encoder. We use Leaky ReLU with slope 0.2 in the discriminator and ReLU activation everywhere else. The architectures of different modules are as follows:

- Style Encoder:

  `c5s1-32,c3s2-64,c3s2-128,avgpool,fc8`

- Generator:

  `c5s1-32-IN,d32-IN,d32-AdaIN,d32-LN,`

  `d32-LN,c5s1-3`

- Discriminator:

  `c5s1-32,c3s2-64,c3s2-128`

The length of the latent style code is set to 8. A style decoder (multi-layer perceptron) has two hidden fully connected layers of 128 filters without normalization, which transforms the latent style code to the parameters of the AdaIN layer.

### 5.4.2 Datasets

We use **MS-Celeb-1M** [2] as our labeled training dataset. As for unlabeled images, we choose **WiderFace** [130] as our training data. WiderFace is dataset collected by retrieving images from search engines with different event keywords. As a face detection dataset, it includes a much wider domain of photos and the faces. Many faces in this dataset still cannot be detected by state-of-the-art detection methods [143]. We only keep the detectable faces in the WiderFace training set as our training data. Our goal is to close the gap between face detection and recognition engine and improve the recognition performance on a general settings with any detectable faces. At the end, we were able to detect about 70K faces from WiderFace, less than 2% of our labeled training data.

To evaluate the face representation models, we test on three benchmarks, namely IJB-B, IJB-C and IJB-S. Although our goal is to improve recognition performance on domains that are different from the training set, we would not like to lose the discrimination power in the original domain

Table 5.1 Ablation study over different training methods of the embedding network. All models has identification loss by default. "DA", "AN", "SM" and "MM" refer to "Domain Alignment", "Augmentation Network", "Single-mode" and "Multi-mode", respectively.

| Method | IJB-C (Vrf) | | | IJB-C (Idt) | | IJB-S (V2S) | | LFW |
|---|---|---|---|---|---|---|---|---|
| | 1e-7 | 1e-6 | 1e-5 | Rank1 | Rank5 | Rank1 | Rank5 | Accuracy |
| Baseline | 62.90 | 82.94 | 90.73 | 94.90 | 96.77 | 53.23 | 62.91 | 99.80 |
| + DA | 72.74 | 85.33 | 90.52 | 94.99 | 96.75 | 56.35 | **66.77** | **99.82** |
| + DA + AN (SM) | 74.80 | 87.58 | **91.94** | 95.51 | 97.09 | 56.98 | 65.66 | 99.80 |
| **+ DA + AN (MM)** | **77.39** | **87.92** | 91.86 | **95.61** | **97.13** | **57.33** | 65.37 | 99.75 |

(high-quality photos) either. Therefore, during ablation we also evaluate our models on the standard **LFW** [3] protocol, which is a celebrity photo dataset, similar to the labeled training data (MS-Celeb-1M). Note that the accuracy on the LFW protocol is highly saturated, so the main goal is just to check whether there is a significant performance drop on the constrained faces while increasing the generalizability to unconstrained ones.

### 5.4.3 Ablation Study

In this section, we conduct an ablation study to quantitatively evaluate the effect of different modules proposed in this chapter. In particular, we have two modules to study: Domain Alignment (DA) and Augmentation Network (AN). The performance is shown in Table 5.1. As we already showed in Fig. 5.3, domain adversarial loss is able to force smaller domain gaps between the sub-domains in WiderFace and the celebrity faces, even though we do not have access to those domain labels. Consequently, we observe the performance improvement on most of the protocols on IJB-C and IJB-S. Introducing the augmentation network (AN) further helps improving the performance on unconstrained benchmarks, where a multi-mode (MM) augmentation network outperforms a single-model (SM) augmentation network.

We also ablate over the training modules of the augmentation network. In particular, we consider to remove the following modules for different variants: Latent-style code for multi-mode generation (MM), Image Discriminator ($D_I$), Reconstruction Loss (Rec), Style Discriminator ($D_z$ ) and the architecture without downsampling (ND). The qualitative results of different models are shown in


Figure 5.6 Ablation study of the augmentation network. Input images are shown in the first column. The subsequent columns show the results of different models trained without a certain module or loss. The texture style codes are randomly sampled from the normal distribution.

Fig. 5.6. Without the latent style code (Model a), the augmentation network can only output one deterministic image for each input, which mainly applies blurring to the input image. Without the image adversarial loss (Model b), the model cannot capture the realistic variations in the unlabeled dataset and the style code can only change the color channel in this case. Without the Reconstruction Loss (Model c), the model is trained only with adversarial loss but without the regularization of content preservation. And therefore, we see clear artifacts on the output images. However, adding reconstruction loss alone hardly helps, since the latent code used in the reconstruction of the unlabeled images could be very different from the prior distribution $p(z)$ that we use for generation. Therefore, similar artifacts can be observed if we do not add latent code adversarial loss (Model d). As for the architecture, if we choose to use an encoder-decoder style network as in the original MUNIT [141], with downsampling and upsampling (Model e), we observe that the output images are always blurred due to the loss of spatial information. In contrast, with our architecture (Model f), the network is capable of augmenting images with diverse color, blurring and illumination styles but without clear artifacts.

Furthermore, we incorporate these different variants of augmentation networks into training and

Table 5.2 Ablation study over different training methods of the augmentation network. "MM", "$D_I$", "$D_Z$", "rec", "ND" refer to "Multi-mode", "Image Discriminator", "Reconstruction Loss", "Latent Style Discriminator" and "No Downsampling", respectively. The first row is a baseline that uses only the domain adversarial loss but no augmentation network. "Model (a)" is a single-mode translation network that does not use latent style code.

| Model | Modules | | | | | IJB-C (Vrf) | | | IJB-C (Idt) | | IJB-S (V2S) | | LFW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MM | $D_I$ | Rec | $D_Z$ | ND | 1e-7 | 1e-6 | 1e-5 | Rank1 | Rank5 | Rank1 | Rank5 | Accuracy |
| | | | | | | 72.74 | 85.33 | 90.52 | 94.99 | 96.75 | 56.35 | 66.77 | 99.82 |
| (a) | | ✓ | | | ✓ | 74.80 | 87.58 | 91.94 | 95.51 | 97.09 | 56.98 | 65.66 | 99.80 |
| (b) | ✓ | | ✓ | ✓ | ✓ | 75.32 | 88.00 | 91.71 | 95.42 | 97.04 | 57.54 | 66.72 | 99.75 |
| (c) | ✓ | ✓ | | | ✓ | 74.51 | 87.49 | 91.97 | 95.61 | 97.18 | 57.17 | 66.24 | 99.78 |
| (d) | ✓ | ✓ | ✓ | | ✓ | 75.07 | 88.11 | 92.19 | 95.66 | 97.12 | 56.85 | 64.87 | 99.78 |
| (e) | ✓ | ✓ | ✓ | ✓ | | 73.99 | 86.52 | 91.33 | 95.33 | 97.04 | 58.47 | 66.00 | 99.73 |
| (f) | ✓ | ✓ | ✓ | ✓ | ✓ | 77.39 | 87.92 | 91.86 | 95.61 | 97.13 | 57.33 | 65.37 | 99.75 |

show the results in Table 5.6. The baseline model here is a model that only uses domain alignment loss without augmentation network. In fact, compared with this baseline, using all different variants of the augmentation network achieves performance improvement in spite of the artifacts in the generated images. But a more stable improvement is observed for the proposed augmentation network across different evaluation protocols. We also show more examples of augmented images in Figure 5.6.

### 5.4.4 Quantity vs. Diversity

Although we have shown in Sec. 5.4.3 that utilizing unlabeled data leads to better performance on challenging testing benchmarks, generally it shall be expected that simply increasing the number of labeled training data can also have a similar effect. Therefore, in this section, we conduct a more detailed study to answer such a question: *which is more important for feature generalizability: quantity or diversity of the training data?* In particular, we train several supervised models by adjusting the number of labeled training data. For each such model, we also train a corresponding model with additional unlabeled data. The evaluation results are shown in Figure 5.7.

On the IJB-S dataset, which is significantly different from the labeled training data, we see that the models trained with unlabeled data consistently outperforms the supervised baselines with a large margin. In particular, the proposed method achieves better performance than the supervised baseline

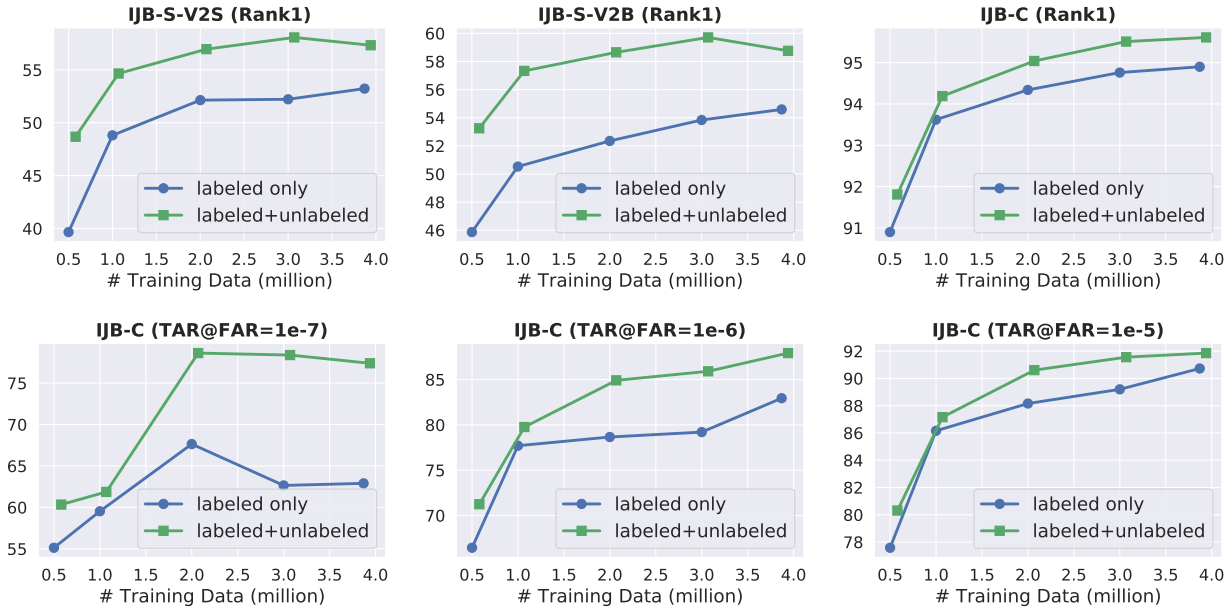



Figure 5.7 Evaluation results on IJB-C and IJB-S with different protocols and different number of labeled training data.

even when there is only one-fourth of the overall labeled training data (1M vs 4M), indicating the value of data diversity during training. Note that there is a significant performance boost when increasing the number of labeled samples from 0.5M to 1M. However, after that, the benefit of acquiring more labeled data plateaus and in fact it is more helpful to introduce 70K unlabeled data than 3M additional labeled data.

On the IJB-C dataset, for both verification and identification protocols, we observe a similar trend as the IJB-S dataset. In particular, larger improvement is achieved at lower FARs. This is because the verification threshold at lower FARs is affected by the low quality test data (difficult impostor pairs), which is more similar to our unlabeled data. Another interesting observation is that the improvement margin increases when there is more labeled data. Note that in general semi-supervised learning, we would expect less improvement by using unlabeled data when there is more labeled data. But it is the opposite in our case because the unlabeled data has different characteristics than the labeled data. So when the performance of supervised model saturates with sufficient labeled data, transferring the knowledge from diverse unlabeled data becomes more helpful.

For both IJB-S and IJB-C (TAR@FAR=1e-7), we observe that after a certain point, adding

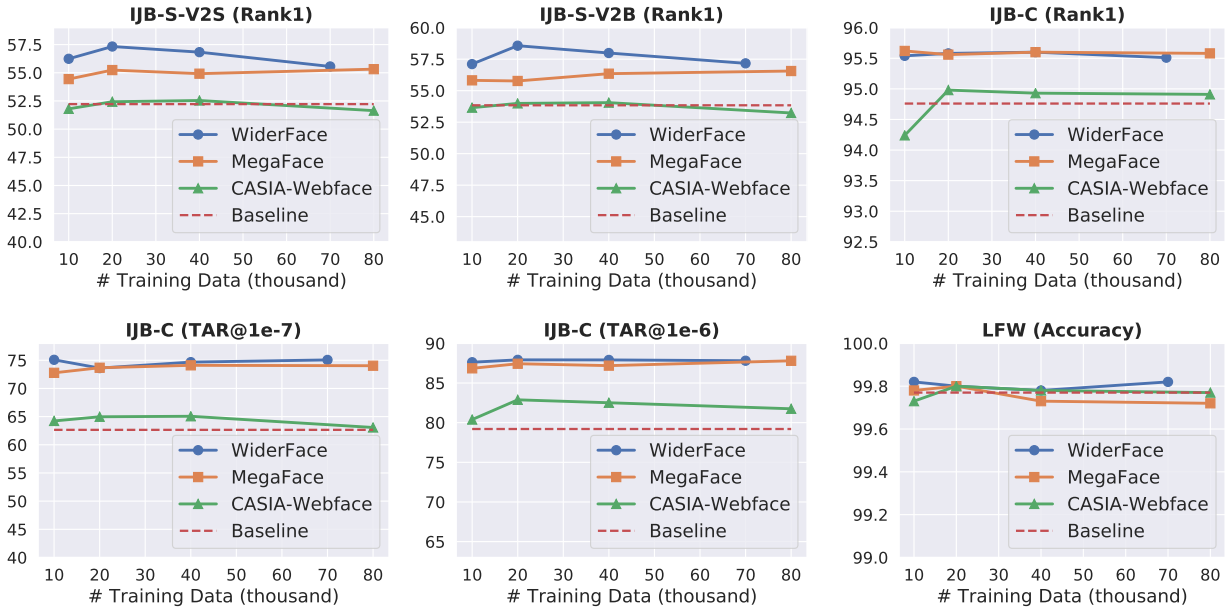



Figure 5.8 Evaluation Results on IJB-S, IJB-C and LFW with different protocols and different number and choice of unlabeled training data. The red line here refers the performance of the supervised baseline which does not use any unlabeled data.

more labeled data does not boost performance any more and the performance starts to fluctuate. This happens because the new labeled data does not necessarily help with those hard cases. Based on these results, we conclude that *when the number of labeled training data is small, it is more important to increase the quantity of the labeled dataset. Once there is sufficient labeled training data, the generalizablity of the representation tends to saturate while the diversity of the training data becomes more important*.

## 5.5 Choice of the Unlabeled Dataset

In Section 5.4.4, we discussed on the impact of the quantity/diversity of training data on feature generalizability, where we conducted the experiments by adjusting the number of labeled faces. Here, we extend the discussion by showing more experiments on the choice of unlabeled dataset. In addition to the WiderFace dataset, we consider to utilize two other datasets: MegaFace [117] and CASIA-WebFace [50]. For MegaFace, we only use the distractor images in their identification

protocol, which are crawled from album photos on Flicker and present a larger degree of variation compared with the faces in MS-Celeb-1M. CASIA-WebFace, similar to MS-Celeb-1M, is mainly composed of celebrity photos, and therefore it should not introduce much additional diversity. Note that CASIA-WebFace is a labeled dataset but we ignore its labels for this experiment. The diversity (facial variation) of the three datasets can ranked as: *WiderFace* > *MegaFace* > *CASIA-WebFace*. For both MegaFace and CASIA-Webface, we choose a random subset to match the number of the WiderFace. Furthermore, to see the impact of the quantity of unlabeled dataset, we also train the models with different numbers of unlabeled data. Then, we evaluate all the models on IJB-S, IJB-C and LFW. The reason to evaluate on LFW here is to see the impact of different unlabeled datasets on the performance in the original domain. The results are shown in Figure 5.8. Note that due to the large number of experiments, we do not use augmentation network here. But empirically we found the trends would be similar.

From Figure 5.8, it can be seen that in general, the more diverse the unlabeled dataset is, the more performance boost it leads to. In particular, using CASIA-WebFace as the unlabeled dataset hardly improves performance on any protocol. This is expected because CASIA-WebFace is very similar to MS-Celeb-1M and hence it cannot introduce additional diversity to regularize the training of face representations. Using MegaFace distractors as the unlabeled dataset improves the performance on both IJB-C and IJB-S, both of which have more variations than the MS-Celeb-1M. Using WiderFace as the unlabeled dataset further improves the performance on the IJB-S dataset. Note that all the models in this experiment maintain the high performance on the LFW dataset. In other words, *using a more diverse unlabeled dataset would not deteriorate the performance on the original domain and safely improves the performance on the challenging new domains*. An additional result that we can observe is that the size of the unlabeled dataset does not have a clear effect compared to its diversity.

### 5.5.1 Comparison with State-of-the-Art FR Methods

In Table 5.3 we show more complete results on IJB-C dataset and compare our method with other state-of-the-art methods. In generally, we observe that with fewer labeled training samples and

Table 5.3 Performance comparison with state-of-the-art methods on the IJB-C dataset.

| Method | Data | Model | Verification | | | | Identification | |
|---|---|---|---|---|---|---|---|---|
| | | | 1e-7 | 1e-6 | 1e-5 | 1e-4 | Rank1 | Rank5 |
| Cao et al. [44] | 13.3M | SE-ResNet-50 | - | - | 76.8 | 86.2 | 91.4 | 95.1 |
| PFE [54] | 4.4M | ResNet-64 | - | - | 89.64 | 93.25 | 95.49 | 97.17 |
| ArcFace [38] | 5.8M | ResNet-50 | 67.40 | 80.52 | 88.36 | 92.52 | 93.26 | 95.33 |
| Ranjan et al. [146] | 5.6M | ResNet-101 | 67.4 | 76.4 | 86.2 | 91.9 | 94.6 | 97.5 |
| AFRN [147] | 3.1M | ResNet-101 | - | - | 88.3 | 93.0 | **95.7** | **97.6** |
| Baseline | 3.9M | ResNet-50 | 62.90 | 82.94 | 90.73 | 94.57 | 94.90 | 96.77 |
| Proposed | 4.0M | ResNet-50 | **77.39** | **87.92** | **91.86** | **94.66** | 95.61 | 97.13 |

Table 5.4 Performance comparison with state-of-the-art methods on the IJB-B dataset.

| Method | Data | Model | Verification | | | | Identification | |
|---|---|---|---|---|---|---|---|---|
| | | | 1e-6 | 1e-5 | 1e-4 | 1e-3 | Rank1 | Rank5 |
| Cao et al. [44] | 13.3M | SE-ResNet-50 | - | 70.5 | 83.1 | 90.8 | 90.2 | 94.6 |
| Comparator [94] | 3.3M | ResNet-50 | - | - | 84.9 | 93.7 | - | - |
| ArcFace [38] | 5.8M | ResNet-50 | 40.77 | 84.28 | 91.66 | 94.81 | 92.95 | 95.60 |
| Ranjan et al. [146] | 5.6M | ResNet-101 | **48.4** | 80.4 | 89.8 | 94.4 | 93.3 | 96.6 |
| AFRN [147] | 3.1M | ResNet-101 | - | 77.1 | 88.5 | 94.9 | **97.3** | **97.6** |
| Baseline | 3.9M | ResNet-50 | 40.12 | 84.38 | **92.79** | **95.90** | 93.85 | 96.55 |
| Proposed | 4.0M | ResNet-50 | 43.38 | **88.19** | 92.78 | 95.86 | 94.62 | 96.72 |

number of parameters, we are able to achieve state-of-the-art performance on most of the protocols. Particularly at low FARs, the proposed method outperforms the baseline methods with a good margin. This is because at a low FAR, the verification threshold is mainly determined by low quality impostor pairs, which are instances of the difficult face samples that we are targeting with additional unlabeled data. Similar trend is observed for IJB-B dataset (Table 5.4). Note that because of fewer number of face pairs, we are only able to test at higher FARs for IJB-B dataset.

In Table 5.5 we show the results on two different protocols of IJB-S. Both the Surveillance-to-Still (V2S) and Surveillance-to-Booking (V2B) protocols use surveillance videos as probes and mugshots as gallery. Therefore, IJB-S results represent a cross domain comparison problem. Overall, the proposed system achieve new state-of-the-art performance on both protocols.

## 5.6 Conclusions

In this chapter, we have proposed a semi-supervised framework of learning robust face representation that could generalize to unconstrained faces beyond the labeled training data. Without collecting

Table 5.5 Performance on the IJB-S benchmark.

| Method | Surveillance-to-Still | | | | | Surveillance-to-Booking | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Rank1 | Rank5 | Rank10 | 1% | 10% | Rank1 | Rank5 | Rank10 | 1% | 10% |
| MARN [148] | 58.14 | 64.11 | - | 21.47 | - | 59.26 | 65.93 | - | 32.07 | - |
| PFE [54] | 50.16 | 58.33 | 62.28 | 31.88 | 35.33 | 53.60 | 61.75 | 62.97 | 35.99 | 39.82 |
| ArcFace[38] | 50.39 | 60.42 | 64.74 | 32.39 | 42.99 | 52.25 | 61.19 | 65.63 | 34.87 | 43.50 |
| Baseline | 53.23 | 62.91 | 67.83 | 31.88 | 43.32 | 54.26 | 64.18 | 69.26 | 32.39 | 44.32 |
| Proposed | **59.29** | **66.91** | **69.63** | **39.92** | **50.49** | **60.58** | **67.70** | **70.63** | **40.80** | **50.31** |

domain specific data, we utilized a relatively small unlabeled dataset containing diverse styles of face images. In order to fully utilize the unlabeled dataset, two methods are proposed. First, we showed that the domain adversarial learning, which is common in adaptation methods, can be applied in our setting to reduce domain gaps between labeled faces and hidden sub-domains. Second, we propose an augmentation network that can capture different visual styles in the unlabeled dataset and apply them to the labeled images during training, making the face representation more discriminative for unconstrained faces. Our experimental results show that as the number of labeled images increases, the performance of the supervised baseline tends to saturate on the challenging testing scenarios. Instead, introducing more diverse training data becomes more important and helpful. In a few challenging protocols, we showed that the proposed method can outperform the supervised baseline with less than half of the labeled data. By training on the labeled MS-Celeb-1M dataset and unlabeled WiderFace dataset, our final model achieves state-of-the-art performance on challenging benchmarks.

# Chapter 6

# Summary

In this thesis, we first review the history of face recognition problem and its solutions. The recognition pipeline includes three steps: normalization, feature learning and similarity metric. We show that existing methods in each step in this pipeline face certain challenges when applied to real-world face recognition scenarios. Thus, four methods are proposed to improve these steps. First, to handle the large pose variation, an attention module is proposed to automatically localize salient facial areas. In contrast to conventional methods to normalize faces by transformation, the proposed method does not explicitly transform the input image. Instead, it automatically discovers salient facial areas and incorporates their information into the global face representation. Second, we propose a new type of face representation, namely probabilistic face embeddings (PFEs). We show that by converting deterministic face embeddings into PFEs, we not only achieve a better interpretability and safety control, but also boost the recognition performance by incorporating data uncertainty into the similarity metric. Third, for the feature extraction, we found that a conventional deep learning framework would suffer from data bias if we simply introduce more variation to augment the training data. Thus, we propose a universal learning framework that decouples the feature embeddings during training to reduce the negative impact of different augmentation on each other. During testing, these decoupled features are combined under the uncertainty framework to handle different types of variations. However, such a learning framework is still limited by

manually designed facial variations, which could be different from data distribution of unconstrained faces in real world applications. Finally, we propose a semi-supervised learning framework, which utilizes an auxiliary unlabeled dataset to regularize the embedding model during training. We use a generative model to automatically discover the latent styles within the unlabeled dataset and transfer them to augment the labeled images. Then we combine the regularization in both the feature and image spaces to build a more generalizable face embedding to boost unconstrained face recognition performance.

## 6.1 Contributions

The main contributions of this thesis are as follows:

1. A spatial transformer-based attention module that automatically detects salient facial regions to extract local features. The attention module could serve as an alternative to complicated normalization techniques to reduce the variations in face images. Further, it could help to discover discriminative local features.

2. A framework that combines multiple region attention modules to extract local features and incorporates them into global facial representation. Experimental results on unconstrained face databases show that the method could effectively boost the performance. And the performance further increases when additional region attention modules are incorporated into the framework.

3. A new type of face representation that takes feature uncertainty into account. We show that deterministic embeddings, which are used in almost all ongoing studies on face recognition, suffer from a feature ambiguity dilemma, which cannot be solved by increasing the model size or augmenting the training data. Instead, we propose to convert pre-trained deep face representations into PFEs by representing each face image as a distribution in the latent space. The probabilistic embedding has a better interpretability and can be used as a quality

assessment method to control the enrollment of face images.

4. A probabilistic framework that could effectively utilize data uncertainty to combine and compare different PFEs to improve the face recognition performance. Evaluation results on unconstrained face recognition benchmarks show that the method consistently improves the recognition performance compared to conventional deterministic embeddings.

5. An universal feature learning framework that learns a set of decoupled face representations. A confidence-controlled face identification loss and a variation-based decoupling loss are proposed in the feature learning process to effectively handle different types of variations in the training data. Experiments show that conventional approaches could suffer from new variations added into training data while the proposed method incrementally enhances the feature representations when additional types of variations are introduced.

6. By combining the universal face representation framework and the PFEs, the proposed method achieves state-of-the-art performance on several challenging recognition benchmarks, including IJB-C, TinyFace and IJB-S.

7. A semi-supervised learning framework for generalizing face representations with unlabeled data, where a representation learning method of joint regularization from both image and feature domains.

8. A multi-mode image translation module is proposed to perform data-driven augmentation to increase the diversity of the labeled training samples.

9. Empirical results show that the regularization of unlabeled data helps to improve the recognition performance on unconstrained testing datasets.

## 6.2 Suggestions for Future Work

Some of the ongoing and possible future directions within the scope of robust unconstrained deep face recognition are as follows:

- **Uncertainty-aware Representation Learning** In Chapter 3, we propose an uncertainty-aware face representation, i.e. PFE, to boost face recognition performance by incorporating uncertainty information into the face comparison process. However, the issue of data uncertainty also exists in the learning process as well. Thus, another direction that is worth exploring is to study whether modeling data uncertainty could accelerate the training of face embeddings.

- **Domain Generalization** In Chapter 4 and Chapter 5, we use manually designed transformations and an unlabeled dataset to generalize supervised models, respectively. Another option is to combine several heterogeneous labeled datasets from different sources to train a more generalizable model. Such a framework is known as Domain Generalization. Although currently we do not have access to large-scale face datasets with clear domain gaps, we believe it would be an interesting research direction to explore if one could collect such datasets.

- **Self-supervised Learning on Unlabeled Data** In Chapter 5, we showed that diversity is more important than the amount of unlabeled data. As such, we believe there is still space in terms of methodology that could further utilize a larger set of unlabeled data to boost the performance. A possible direction is to apply self-supervised learning to unlabeled data, which has recently been shown successful on image classification tasks.

# APPENDIX

# PUBLICATIONS


[1] D. Deb, S. Wiper, S. Gong, Y. Shi, C. Tymoszek, A. Fletcher, and A. K. Jain, "Face recognition: Primates in the wild," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019.

[2] Y. Shi, C. Otto, and A. K. Jain, "Face clustering: representation and pairwise constraints," *IEEE Transactions on Information Forensics and Security*, 2018.

[3] Y. Shi and A. Jain, "Improving face recognition by exploring local features with visual attention," in *ICB*, 2018.

[4] Y. Shi and A. K. Jain, "Docface: Matching id document photos to selfies," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019.

[5] Y. Shi and A. K. Jain, "Docface+: Id document to selfie matching," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.

[6] Y. Shi, D. Deb, and A. K. Jain, "Warpgan: Automatic caricature generation," in *CVPR*, 2019.

[7] S. Gong, Y. Shi, N. D. Kalka, and A. K. Jain, "Video face recognition: Component-wise feature aggregation network (c-fan)," in *ICB*, 2019.

[8] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *ICCV*, 2019.

[9] S. Gong, Y. Shi, and A. Jain, "Low quality video face recognition: Multi-mode aggregation recurrent network (marn)," in *CVPR Workshops*, 2019.

[10] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, "Towards universal representation learning for deep face recognition," in *CVPR*, 2020.

[11] Y. Shi and A. K. Jain, "Boosting unconstrained face recognition with auxiliary unlabeled data," in *CVPR Workshops*, 2021.

[12] Y. Shi, D. Aggarwal, and A. K. Jain, "Lifting 2d stylegan for 3d-aware face generation," in *CVPR*, 2021.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


[1] N. D. Kalka, B. Maze, J. A. Duncan, K. J. O'Connor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain, "IJB-S : IARPA Janus Surveillance Video Benchmark ," in *BTAS*, 2018.

[2] 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.

[3] 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," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[4] 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.

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

[6] Z. Cheng, X. Zhu, and S. Gong, "Low-resolution face recognition," in *ACCV*, 2018.

[7] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, 2016.

[8] S. Liao, Z. Lei, D. Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," in *IJCB*, 2014.

[9] "Lax trials facial recognition and advanced imaging technology at security." `https://www.futuretravelexperience.com/2018/02/lax-trialling-facial-recognition-and-advanced-imaging-technology-at-security/`, 2018.

[10] "iphone's face id isn't perfect, but you can make it better." `https://www.cnet.com/how-to/iphones-face-id-problems-tricks-tips/`, 2018.

[11] "China's alipay adds sought-after beauty filters to face-scan payments." `https://anith.com/chinas-alipay-adds-sought-after-beauty-filters-to-face-scan-payments-techcrunch/`, 2019.

[12] "Please run australia's facial recognition surveillance system on the ato san." `https://www.zdnet.com/article/please-run-australias-facial-recognition-surveillance-system-on-the-ato-san/`, 2019.

[13] "How facial recognition is taking over airports." `https://www.cnn.com/travel/article/airports-facial-recognition/index.html`, 2019.

[14] "How facial recognition is fighting child sex trafficking." https://www.wired.com/story/how-facial-recognition-fighting-child-sex-trafficking, 2019.

[15] "One billion surveillance cameras will be watching around the world in 2021, a new study says." https://www.cnbc.com/2019/12/06/one-billion-surveillance-cameras-will-be-watching-globally-in-2021.html, 2019.

[16] T. Kanade, *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, 1973.

[17] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, 1995.

[18] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *CVPR*, 1991.

[19] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. on PAMI*, vol. 19, no. 7, 1997.

[20] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[21] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. on PAMI*, vol. 28, 2006.

[22] 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.

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

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[26] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. on PAMI*, 2017.

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

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

[29] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *NIPS*, 2014.

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

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

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

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

[34] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017.

[35] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *arXiv:1801.05599*, 2018.

[36] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018.

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

[38] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CVPR*, 2019.

[39] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Generative models for recognition under variable pose and illumination," in *FG*, 2000.

[40] "Iarpa janus program." `https://www.iarpa.gov/index.php/research-programs/janus`.

[41] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, *et al.*, "Iarpa janus benchmark-b face dataset," in *CVPR Workshops*, 2017.

[42] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *ICB*, 2018.

[43] D. Wang, C. Otto, and A. K. Jain, "Face search at scale," *IEEE Trans. on PAMI*, vol. 39, 2016.

[44] 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.

[45] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," in *WACV*, 2016.

[46] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: $l_2$ hypersphere embedding for face verification," *ACM MM*, 2017.

[47] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, 2017.

[48] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. on Image Processing*, 2017.

[49] J. Zhao, L. Xiong, Y. Cheng, Y. Cheng, J. Li, L. Zhou, Y. Xu, J. Karlekar, S. Pranata, S. Shen, *et al.*, "3d-aided deep pose-invariant face recognition.," in *IJCAI*, p. 11, 2018.

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

[51] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR*, 2011.

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

[53] Y. Shi and A. Jain, "Improving face recognition by exploring local features with visual attention," in *ICB*, 2018.

[54] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *ICCV*, 2019.

[55] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, "Towards universal representation learning for deep face recognition," in *CVPR*, 2020.

[56] Y. Shi and A. K. Jain, "Boosting unconstrained face recognition with auxiliary unlabeled data," in *CVPR Workshops*, 2021.

[57] O. M. Parkhi, A. Vedaldi, A. Zisserman, *et al.*, "Deep face recognition.," in *BMVC*, 2015.

[58] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. on PAMI*, vol. 32, no. 9, 2010.

[59] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *ECCV*, 2014.

[60] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, 2014.

[61] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *ICCV*, 2015.

[62] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv:1412.7755*, 2014.

[63] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.

[64] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, 2017.

[65] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *NIPS*, 2015.

[66] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *CVPR*, 2015.

[67] Y. Zhong, J. Chen, and B. Huang, "Toward end-to-end face recognition through alignment learning," *IEEE Signal Processing Letters*, vol. 24, no. 8, 2017.

[68] A. Hasnat, J. Bohné, J. Milgram, S. Gentric, and L. Chen, "Deepvisage: Making face recognition simple yet with powerful generalization skills," in *ICCV*, 2017.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

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

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

[72] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *BMVC*, 2015.

[73] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016.

[74] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *NIPS*, 2017.

[75] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Computation*, 1992.

[76] R. M. Neal, *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

[77] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2013.

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

[79] S. Khan, M. Hayat, W. Zamir, J. Shen, and L. Shao, "Striking the right balance with uncertainty," *arXiv:1901.07590*, 2019.

[80] U. Zafar, M. Ghafoor, T. Zia, G. Ahmed, A. Latif, K. R. Malik, and A. M. Sharif, "Face recognition with bayesian convolutional networks for robust surveillance systems," *EURASIP Journal on Image and Video Processing*, 2019.

[81] Y. Xu, X. Fang, X. Li, J. Yang, J. You, H. Liu, and S. Teng, "Data uncertainty in face recognition," *IEEE Trans. on Cybernetics*, 2014.

[82] G. Shakhnarovich, J. W. Fisher, and T. Darrell, "Face recognition from long-term observations," in *ECCV*, 2002.

[83] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face recognition with image sets using manifold density divergence," in *CVPR*, 2005.

[84] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in *CVPR*, 2010.

[85] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *ICML*, 2015.

[86] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *CVPR*, 2013.

[87] P. Hiremath, A. Danti, and C. Prabhakar, "Modelling uncertainty in representation of facial features for face recognition," in *Face recognition*, 2007.

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

[89] Y. Liu, J. Yan, and W. Ouyang, "Quality aware network for set to set recognition," in *CVPR*, 2017.

[90] W. Xie and A. Zisserman, "Multicolumn networks for face recognition," in *ECCV*, 2018.

[91] S. Gong, Y. Shi, and A. K. Jain, "Video face recognition: Component-wise feature aggregation network (c-fan)," in *ICB*, 2019.

[92] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. on Image Processing*, 2018.

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

[94] W. Xie, L. Shen, and A. Zisserman, "Comparator networks," in *ECCV*, 2018.

[95] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Trans. on Information Forensics and Security*, 2015.

[96] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker, "Unsupervised domain adaptation for distance metric learning," in *CVPR*, 2019.

[97] K. Sohn, W. Shang, X. Yu, and M. Chandraker, "Unsupervised domain adaptation for distance metric learning," in *ICLR*, 2019.

[98] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?," in *ECCV*, 2016.

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

[100] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *CVPR*, 2019.

[101] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," in *BTAS*, 2016.

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

[103] H. Bilen and A. Vedaldi, "Universal representations: The missing link between faces, text, planktons, and cat breeds," *arXiv:1701.07275*, 2017.

[104] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *NIPS*, 2017.

[105] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *CVPR*, 2018.

[106] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos, "Towards universal object detection by domain attention," in *CVPR*, 2019.

[107] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *ECCV*, 2012.

[108] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *ICML*, 2013.

[109] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, pp. 5542–5550, 2017.

[110] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI*, 2018.

[111] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. E. A. Seddik, and M. Tamaazousti, "Learning more universal representations for transfer-learning," *IEEE Trans. on PAMI*, 2019.

[112] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[113] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017.

[114] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[115] X. Yu, F. Zhou, and M. Chandraker, "Deep deformation network for object landmark localization," in *ECCV*, 2016.

[116] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," in *ECCV*, 2018.

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

[118] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *CIP*, 2014.

[119] J. Zhao, L. Xiong, J. Li, J. Xing, S. Yan, and J. Feng, "3d-aided dual-agent gans for unconstrained face recognition," *IEEE Trans. on PAMI*, 2018.

[120] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *ICCV*, 2015.

[121] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *ICCV*, 2017.

[122] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *CVPR*, 2018.

[123] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *CVPR*, 2019.

[124] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. on Neural Networks*, 2010.

[125] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015.

[126] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2017.

[127] K. Sohn, W. Shang, X. Yu, and M. Chandraker, "Unsupervised domain adaptation for face recognition in unlabeled videos," in *CVPR*, 2017.

[128] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *CVPR*, 2018.

[129] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019.

[130] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *CVPR*, 2016.

[131] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICML Workshop*, 2013.

[132] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *NeurIPS*, 2015.

[133] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2017.

[134] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[135] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv:1904.12848*, 2019.

[136] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *ICCV*, 2019.

[137] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *NeurIPS*, 2019.

[138] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv:2001.07685*, 2020.

[139] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[140] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018.

[141] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[142] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization.," in *ICCV*, 2017.

[143] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[144] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv:1607.08022*, 2016.

[145] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[146] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu, A. Nanduri, J.-C. Chen, C. D. Castillo, and R. Chellappa, "A fast and accurate system for face detection, identification, and verification," *IEEE Trans. on Biometrics, Behavior, and Identity Science*, 2019.

[147] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, "Attentional feature-pair relation networks for accurate face recognition," in *ICCV*, 2019.

[148] S. Gong, Y. Shi, and A. Jain, "Low quality video face recognition: Multi-mode aggregation recurrent network (marn)," in *ICCV Workshops*, 2019.