DocFace+: ID Document to Selfie* Matching

Yichun Shi, *Student Member, IEEE,* and Anil K. Jain, *Life Fellow, IEEE*

Abstract—Numerous activities in our daily life require us to verify who we are by showing our ID documents containing face images, such as passports and driver licenses, to human operators. However, this process is slow, labor intensive and unreliable. As such, an automated system for matching ID document photos to live face images (selfies) in real time and with high accuracy is required. In this paper, we propose DocFace+ to meet this objective. We first show that gradient-based optimization methods converge slowly (due to the underfitting of classifier weights) when many classes have very few samples, a characteristic of existing ID-selfie datasets. To overcome this shortcoming, we propose a method, called dynamic weight imprinting (DWI), to update the classifier weights, which allows faster convergence and more generalizable representations. Next, a pair of sibling networks with partially shared parameters are trained to learn a unified face representation with domain-specific parameters. Cross-validation on an ID-selfie dataset shows that while a publicly available general face matcher (InsightFace) only achieves a True Accept Rate (TAR) of $88.78 \pm 1.30\%$ at a False Accept Rate (FAR) of 0.01% on the problem, DocFace+ improves the TAR to $95.95 \pm 0.54\%$.

Index Terms—ID-selfie face matching, face recognition, face verification, access control, document photo, selfies

1 Introduction

Identity verification plays an important role in our daily lives. For example, access control, physical security and international border crossing require us to verify our access (security) level and our identities. A practical and common approach to this problem involves comparing an individual's live face to the face image found in his/her ID document. For example, immigration and customs officials look at the passport photo to confirm a traveler's identity. Clerks at supermarkets in the United States look at the customer's face and driver license to check his/her age when the customer is purchasing alcohol. Instances of ID document photo matching can be found in numerous scenarios. However, it is primarily conducted by humans manually, which is time consuming, costly, and prone to operator errors. A study pertaining to the passport officers in Sydney, Australia, shows that even the trained officers perform poorly in matching unfamiliar faces to passport photos, with a 14% false acceptance rate [1]. Therefore, an accurate and automated system for efficient matching of ID document photos to selfies* is required. In addition, automated ID-selfie matching systems also enable remote authentication applications that are otherwise not feasible, such as onboarding new customers in a mobile app (by verifying their identities for account creation), or account recovery in the case of forgotten passwords. One application scenario of our ID-selfie matching system (DocFace+) is illustrated in Figure 1.

A number of automated ID-selfie matching systems have been deployed at international border crossings. Deployed in 2007, SmartGate [2] in Australia (See Figure 2) is the earliest of its kind. Due to an increasing number of travelers to Australia, the Australian government introduced SmartGate at most of its international airports as an electronic passport check for ePassport holders. To use the SmartGate, travelers only need to let a machine read their ePassport chips containing their digital photos and then capture their face images using a camera mounted at the SmartGate. After verifying a traveler's identity by face comparison, the gate is automatically opened for the traveler to enter Australia. Similar machines have also been installed in the UK (ePassport gates) [3], USA (US Automated Passport Control) [4] and other countries. In China, such verification systems have been deployed at various locations, including train stations, for matching Chinese ID cards with live faces [5]. In addition to international border control, some businesses [6], [7] are utilizing face recognition solutions to ID document verification for online services.

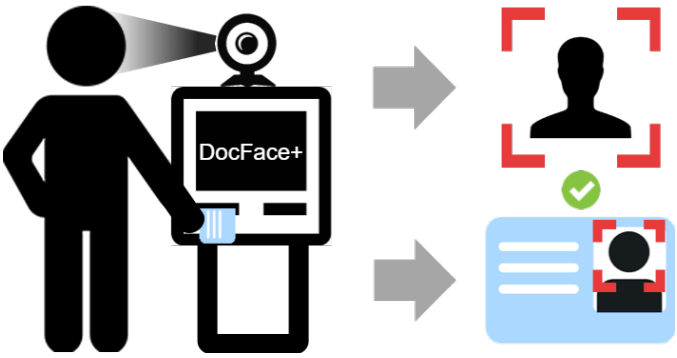


Fig. 1: An application scenario of the DocFace+ system. The kiosk scans the ID document photos or reads the photo from the embedded chip and the camera takes another photo of the holder's live face (selfie). By comparing the two photos, the system decides whether the holder is indeed the owner of the ID document.

The problem of ID-selfie matching poses numerous challenges that are different from general face recognition. For typical unconstrained face recognition tasks, the main challenges are due to pose, illumination and expression (PIE) variations. On the other hand, in ID-selfie matching, we are comparing a scanned or digital document photo to a digital camera photo of a live face. Assuming that the user is cooperative, both of the images are captured under constrained conditions and large PIE variations would not be present. However, (1) the low quality of document

* *Technically, the word "selfies" refers to self-captured photos from mobile phones. But here, we define "selfies" as any self-captured live face photos, including those from mobile phones and kiosks.*

• *Y. Shi and A. K. Jain are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824. E-mail: shiyichu@msu.edu, jain@cse.msu.edu*

(a) SmartGate (Australia) [2] (b) ePassport gates (UK) [3] (c) Automated Passport Control (US) [4] (d) ID card gates (China) [5]

Fig. 2: Examples of automatic ID document photo matching systems at international borders.

(a) General face matching

(b) ID-selfie matching

Fig. 3: Example images from (a) LFW dataset [8] and (b) ID-selfie dataset. Each row shows two pairs from each dataset, respectively. Compared with the general unconstrained face recognition shown in (a), ID Document photo matching in (b) does not need to consider large pose variations. Instead, it involves some other challenges such as aging and information loss via image compression.

photos due to image compression[1] and (2) the large time gap between the document issue date and the verification date remain as the primary difficulties (See Figure 3). In addition, since state-of-the-art face recognition systems are based on deep networks, another issue faced in our problem is the lack of a large training dataset (pairs of ID photos and selfies).

In spite of numerous applications and associated challenges, there is a paucity of research on ID-selfie matching. Most of the published studies are now dated [9], [10], [11], [12]. It is important to note that face recognition technology has made tremendous strides in the past five years, mainly due to the availability of large face datasets and the progress in deep neural network architectures. Hence, the earlier published results on ID-selfie matching are now obsolete. To the best of our knowledge, our prior work [13] is the first to investigate the application of deep CNN to this problem, concurrent with Zhu et al. [14].

In this paper, we first briefly review existing studies on the ID-selfie matching problem and other studies related to our work. We then extend our prior work of DocFace [13] to a more robust and accurate method, DocFace+, for building ID-selfie matching systems. We use a large private dataset of Chinese Identity Cards with corresponding selfies[2] to develop the system and to evaluate the performance of (i) two Commercial-Off-The-Shelf (COTS) face matchers, (ii) open-source deep network face matchers, and (iii) the proposed method. We also compare the proposed system on the open benchmark established by Zhu et al [14]. The contributions of the paper are summarized below:

- A new optimization method for classification-based embedding learning on shallow datasets[3].
- A new recognition system containing a pair of partially shared networks for learning unified representations from ID-selfie pairs.
- An evaluation of COTS and public-domain face matchers showing ID-selfie matching is a non-trivial problem with different challenges from general face matching.
- An open-source face matcher[4], namely DocFace+, for ID-selfie matching, which significantly improves the performance of state-of-the-art general face matchers. Our experiment results show that while the publicly available CNN matcher (InsightFace) only achieves a True Accept Rate (TAR) of $88.78 \pm 1.30\%$ at False Accept Rate (FAR) of 0.01% on the problem, DocFace+ improves the TAR to $95.95 \pm 0.54\%$.

2 Related Work

2.1 ID Document Photo Matching

To the best of our knowledge, the first study on ID-selfie matching is attributed to Starovoitov et al. [9], [12]. Assuming all face images are frontal faces without large expression variations, the authors first localize the eyes with Hough Transform. Based on eye locations, the face region is cropped and gradient maps are computed as feature maps. The algorithm is similar to a general constrained face matcher, except it is developed for a document photo dataset. Bourlai et al. [10], [11] considered ID-selfie matching as a comparison between degraded face images, i.e. scanned document photos, and high quality live face images. To eliminate the degradation caused by scanning, Bourlai et al. inserted an image restoration phase before comparing the photos using a general face matcher. In particular, they train a classifier to classify the degradation type for a given image, and then apply

1. Most chips in e-Passports have a memory ranging from 8KB to 30KB; the face images need to be compressed to be stored in the chip. See https://www.readid.com/blog/face-images-in-ePassports

2. This dataset consists of $53,591$ different ID-selfie pairs captured at different locations and at different times. Due to privacy reasons, we cannot release this data in the public domain. To our knowledge, there is no such dataset available in public domain.

3. Face recognition datasets are often described in terms of breadth and depth [15], where breadth refers to the number of classes (subjects) and depth means the average number of samples per class.

4. The source code is available at https://github.com/seasonSH/DocFace

degradation-specific filters to restore the degraded images. Compared with their work on scanned documents, the document photos in our datasets are read from the chips embedded in the Chinese ID Cards. Additionally, our method is not designed for any specific degradation type and could be applied to any type of ID document photos. Concurrent with our prior work [13], Zhu et al. [14] also worked on deep CNN-based ID-selfie matching systems. With 2.5M ID-selfie pairs, also from a private Chinese ID card dataset, they formulated it as a bisample learning problem and proposed to train the network in three stages: (1) pre-learning (classification) on general face datasets, (2) transfer learning (verification), and (3) fine-grained learning (classification). Our work, on the other hand, proposes a special optimization method to address the slow convergence problem of classification-based embedding learning methods on ID-selfie datasets, which does not require multi-stage training. Compared with our prior work [13], the differences in this version are as follows: (1) a larger ID-selfie dataset (over 50,000 subjects), which is a combination of the two small private datasets in [13] and another larger ID-selfie dataset, (2) a different loss function, namely DIAM-Softmax, to learn the face representation, (3) more comprehensive experiments to analyze the effect of each module and (4) an evaluation of the proposed system as well as other face matchers on a new ID-selfie benchmark, Public IvS, released by Zhu et al. in [14].

2.2 Deep Face Recognition

With the success of deep neural networks in the ImageNet competition [16], virtually all of the ongoing research in face recognition now utilizes deep neural networks to learn face representation [17], [18], [19], [20], [21], [22], [23]. Taigman et al. [17] first proposed the application of deep neural networks to learn face representation. They designed an 8-layer convolutional neural network and trained it with a Softmax loss function, a standard loss function for classification problems. They used the outputs of the bottleneck layer as face representation and achieved state-of-the-art performance at that time in 2014. Considering that Softmax loss only encourages large inter-class variations but does not constrain intra-class variations, Sun et al. [18] later proposed to train networks with both a classification signal (Softmax loss) and a metric learning signal (Contrastive loss). About the same time, Schroff et al. [19] proposed a metric learning loss function, named triplet loss, boosting the state-of-the-art performance on the standard LFW protocol [8]. Liu et al. [21], [24], [25] first bridged the gap between classification loss functions and metric learning methods with the Angular-Softmax (A-Softmax) loss function, a modified Softmax loss that classifies samples based on angular distances. Wang et al. [23] recently proposed the Additive Margin Softmax (AM-Softmax), which learns to increase the angular discriminability with an additive margin and is shown to be more robust than A-Softmax.

2.3 Heterogeneous Face Recognition

Heterogeneous face recognition (HFR) is an emerging topic that has become popular in the past few years [26]. It usually refers to face recognition between two different modalities, including visible spectrum images (VIS), near infrared images (NIR) [27], thermal infrared images [28], composite sketches [29], etc. ID-selfie matching can be considered to be a special case of HFR since the images to be matched come from two different sources. Therefore, the techniques used in HFR could be potentially helpful for the ID-selfie problem. Most methods for HFR can be categorized into two types: synthesis-based methods and discriminant feature-based methods. Synthesis-based methods aim to transform the images from one modality into another so that general intra-modality face recognition systems can be applied [29], [30], [31], [32]. On the contrary, discriminant feature-based methods either manually design a modality-invariant visual descriptor or learn a set of features from the training data so that images from different modalities can be mapped into a shared feature space [33], [34], [35]. Recent studies on HFR have focused on utilizing deep neural networks to learn such modality-invariant features. In [36], Liu et al. first proposed to apply deep VIS face features to VIS-NIR matching via transfer learning where their network is fine-tuned on VIS-NIR dataset with a modified triplet loss. He et al. proposed [37] to use a shared convolutional neural network to map VIS and NIR images into three feature vectors: one set of shared features and two modality-specific feature sets, which are then concatenated and trained with three Softmax loss functions. Wu et al. [38] proposed to learn a unified feature space with two correlated modality-specific Softmax loss functions. The weights of the two Softmax loss functions are regularized by a trace norm and a block-diagonal prior to encourage correlation between representations from different modalities and to avoid overfitting.

2.4 Low-shot Learning

Another field related to our work is the low-shot learning problem. In low-shot learning [39], [40], [41], a model is trained in such a way that it is able to generalize to unseen classes, which may have only a few samples. There are two training phases in low-shot learning: the model, or learner, is first trained on a larger classification dataset, and then in testing, a few labeled samples of new classes are given and the model is required to learn a new classifier given these classes. The adjective "low-shot" refers to a small number of images per class. This problem has been receiving growing interest from the machine learning community because humans are very good at adapting to new types of objects (classes) while conventional deep learning methods require abundant samples for discriminating a specific class from others. This is related to the ID-selfie problem since most of the identities only have a few samples, resulting in a shallow dataset. Many methods have been proposed for low-shot learning problem. Koch et al. [39] proposed a simple yet effective approach by learning metrics via siamese network [42] for one-shot recognition. Vinyals et al. [40] proposed the Matching Net where they simulate the testing scenario in the training phase by learning low-shot recognition in mini-batches. This idea was then generalized as *meta-learning*, where an extra meta-learner can learn how to optimize or produce new classifiers [43], [44]. The Prototypical Network by Snell et al. [41] is more relevant to our work. They proposed to learn a network such that prototypes, i.e. average feature vector of an unseen class, can be used for classification. Qi et al. [45], based on the idea of the prototypes, or proxies [46], proposed to imprint the weights of a new classifier with extracted features. We note that their work differs from ours as they utilized the imprinted weights simply as initialization while we use weight imprinting as an optimization method throughout training by dynamically imprinting the weights.

(a) MS-Celeb-1M (b) Private ID-selfie (c) Public IvS

Fig. 4: Example images in each dataset. The left image in each pair in (b) and (c) is the ID photo and on its right is one of its corresponding selfies.

3 Datasets

In this section, we briefly introduce the datasets that are used in this paper. Some example images of the datasets are shown in Figure 4. As stated earlier, due to our NDA privacy issues, we cannot release the Private ID-selfie dataset. But by comparing our results with public face matchers, we believe it is sufficient to show the difficulty of the problem and advantages of the proposed method.

3.1 MS-Celeb-1M

The MS-Celeb-1M dataset [47] is a public domain face dataset facilitating training of deep networks for face recognition. It contains $8,456,240$ face images of $99,892$ subjects (mostly celebrities) downloaded from internet. In our transfer learning framework, it is used to train a base network. Because the dataset is known to have many mislabels, we use a cleaned version[5] of MS-Celeb-1M with $5,041,527$ images of $98,687$ subjects. Some example images from this dataset are shown in Figure 4(a).

3.2 Private ID-selfie

During the experiments, we use a private dataset to develop and evaluate our ID-selfie matching system. It is a combination of a larger ID-selfie dataset and two other smaller datasets used in our prior work [13]. The dataset contains $116,914$ images of $53,591$ identities in all. Each identity has only one ID card photo. A subset of $53,054$ identities have only one selfie while the other 537 have multiple selfies. The ID card photos are read from chips in the Chinese Resident Identity Cards[6]. In our experiments, similar to LFW [8] protocol, we define two views: (1) a development view for hyper-parameter tuning; (2) an evaluation view for analyzing the performance. A 5-fold cross-validation is conducted in the evaluation view. Some example pairs from this dataset are shown in Figure 4(b).

3.3 Public IvS

Public IvS is a dataset released by Zhu et al. [14] for evaluation of ID-selfie matching systems. The dataset is constructed by collecting ID photos and live face photos of Chinese personalities from Internet. The dataset contains $1,262$ identities and $5,503$ images in total. Each identity has one ID photo and 1 to 10 selfies. It is not strictly an ID-selfie dataset since its ID photos are not from real ID cards but are simulated with highly constrained frontal photos. The results on this dataset were shown to be consistent with real-world ID-selfie datasets [14]. However, our experiments show that this dataset is similar to a general face dataset, such as LFW (See Section 5.8). Some example pairs of this dataset are shown in Figure 4(c).

5. https://github.com/AlfredXiangWu/face_verification_experiment.

6. The second-generation Chinese ID cards, which were first launched in 2004 and completely replaced the first generation in 2014, contain an IC chip. The chip stores a compressed face photo of the owner. See more at https://en.wikipedia.org/wiki/Resident_Identity_Card

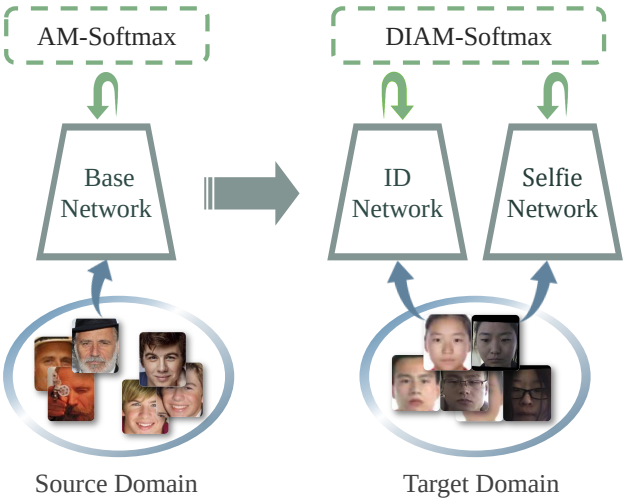


Fig. 5: Work flow of the proposed method. We first train a base model on a large scale unconstrained face dataset. Then, the parameters are transferred to a pair of sibling networks, which share high-level modules. Random ID-selfie pairs are sampled from different classes to train the networks. The proposed DIAM-Softmax is used to learn a shared feature space for both domains of ID and selfie.

4 Methodology

4.1 Overview

In our work, we first train a network as *base model* on a large-scale unconstrained face dataset, i.e. MS-Celeb 1M and then transfer its features to our target domain of ID-selfie pairs. To ensure the performance of transfer learning, we utilize the popular Face-ResNet (DeepVisage) architecture [22] to build the convolutional neural network. For training the base model, we adopt state-of-the-art

Additive Margin Softmax (AM-Softmax) loss function [23] [48]. Then we propose a novel optimization method called dynamic weight imprinting (DWI) to update the weight matrix in AM-softmax when training on the ID-selfie dataset. A pair of sibling networks is proposed for learning domain-specific features of IDs and selfies, respectively, with shared high-level parameters. An overview of the work flow is shown in Figure 5.

4.2 Original AM-Softmax

We use the original *Additive Margin Softmax* (AM-Softmax) loss function [23] [48] for training the base model. Here, we give a short review to gain more understanding of this loss function before we introduce our modifications in the next section. Similar to Angular Softmax [21] and L2-Softmax [49], AM-Softmax is a classification-based loss function for embedding learning, which aims to maximize inter-subject separation and to minimize intra-subject variations. Let $X^s = \{(x_i, y_i)|i = 1, 2, 3, \cdots, N\}$ be our training dataset and $\mathcal{F} : \mathbb{R}^{h\times w\times c} \rightarrow \mathbb{R}^d$ be a feature extraction network, where $x_i \in \mathbb{R}^{h\times w\times c}$ is a face image, y_i is the label and h,w,c are the height, width and number of channels of the input images, respectively. N is the number of training images and d is the number of feature dimensions. For a training sample x_i in a mini-batch, the loss function is given by:

$$\mathcal{L} = -\log p_{y_i}^{(i)} \tag{1}$$

where

$$p_j^{(i)} = \frac{\exp(a_j^{(i)})}{\sum_k \exp(a_k^{(i)})}$$

$$a_j^{(i)} = \begin{cases} s\,\mathbf{w_j}^T\mathbf{f_i} - m, & \text{if } j = y_i \\ s\,\mathbf{w_j}^T\mathbf{f_i}, & \text{otherwise} \end{cases}$$

$$\mathbf{w_j} = \frac{\mathbf{w_j}^*}{\|\,\mathbf{w_j}^*\,\|_2}$$

$$\mathbf{f_i} = \frac{\mathcal{F}(x_i)}{\|\,\mathcal{F}(x_i)\|_2}.$$

Here $\mathbf{w_j}^* \in \mathbb{R}^d$ is the weight vector for j^{th} class, m is a hyper-parameter for controlling the margin and s is a scale parameter. Notice that this formula is a little different from the original AM-Softmax [23] in the sense that the margin m is not multiplied by s, which allows us to automatically learn the parameter s [50]. During training, the loss in Equation (1) is averaged across all images in the mini-batch. The key difference between AM-Softmax and original Softmax is that both the features f_i and the weight vectors w_i are normalized and lie in a spherical embedding space. Thus, instead of classifying samples based on inner products, AM-Softmax, during training, aims to learn features that are separable using cosine similarity, the same metric used in the testing phase. This closes the gap between training and testing as well as the gap between classification learning and metric learning. The normalized weight vector $\mathbf{w_j}$ is considered to be an "agent" or "proxy" of the j^{th} class, representing the distribution of this class in the embedding space [50], [51]. In original AM-Softmax as well as other related works [50], [49], [51], [23], classifier weights are also optimized with stochastic gradient descent (SGD). For the weights $\mathbf{w}_{y_i}$ of the ground-truth class, the gradient

$$\frac{\partial\,\mathcal{L}}{\partial \mathbf{w}_{y_i}} = s(1 - p_{y_i}^{(i)})\,\mathbf{f_i} \tag{2}$$

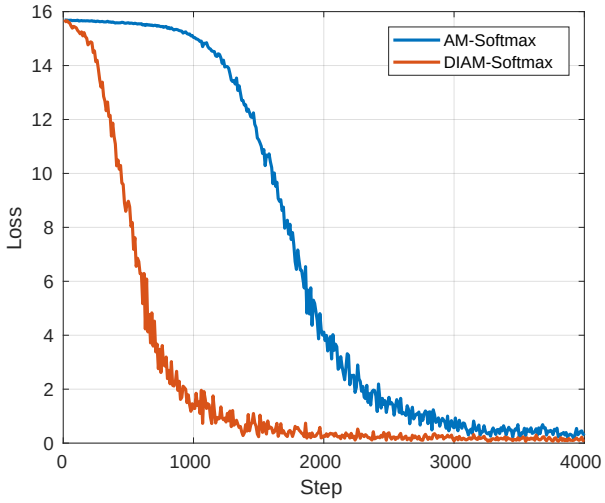
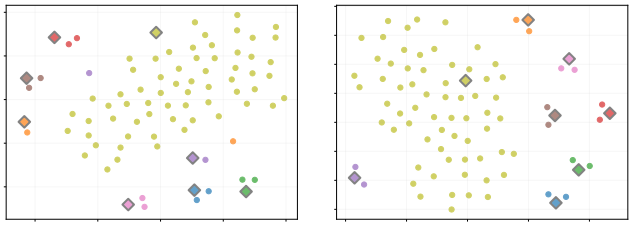


Fig. 6: Training loss of AM-Softmax compared with DIAM-Softmax. DIAM-Softmax shares the same formula as AM-Softmax, but its weights are updated with the proposed DWI instead of SGD.



(a) AM-Softmax (b) DIAM-Softmax

Fig. 7: Visualization of 8 randomly selected classes in the training dataset. The dimensionality of the original embedding space is 512 and is reduced to 2 with t-SNE [52]. The circles are training samples and the diamonds are normalized classifier weights. In original AM-Softmax, many weight vectors shift from their corresponding distributions when the dataset is shallow and update signals are sparse, leading to a low convergence speed. In DIAM-Softmax, the classifier weights are better aligned with the respective class samples.

serves as an attraction signal to pull $\mathbf{w}_{y_i}$ closer to $\mathbf{f_i}$. For other classes $j \neq y_i$, the gradient

$$\frac{\partial\,\mathcal{L}}{\partial\,\mathbf{w_j}} = -sp_j^{(i)}\,\mathbf{f_i} \tag{3}$$

provides a repulsion signal to push $\mathbf{w_j}$ away from $\mathbf{f_i}$. Compared with metric learning methods [42] [19], which update the network based only on the statistics of stochastic mini-batches and need to mine non-trivial pairs/triplets for faster convergence, the normalized classifier weights allows AM-Softmax to capture global distributions of different classes in the embedding space to accelerate training.

4.3 Dynamic Weight Imprinting

In spite of the success of AM-Softmax and other classification-based embedding learning loss functions on general face recognition [49], [50], [21], we found them to be less competitive for transfer learning on the ID-selfie dataset [13]. In fact, it is often the case that they converge very slowly and get stuck at a poor local minimum. As shown in Figure 6, the original AM-Softmax does not start to converge after several epochs[7]. To gain more insight

7. In this example, each epoch is approximately 300 steps.

TABLE 1: The mean (and s.d.) of performance of AM-Softmax and DIAM-Softmax on the Private ID-selfie dataset based on 5-fold cross-validation. Compared with the proposed method, original AM-Softmax takes twice as long to converge and overfits with more training.

Method	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
AM-Softmax	91.65 ± 1.19	95.13 ± 0.72	97.08 ± 0.45
AM-Softmax (2x steps)	92.53 ± 1.09	95.57 ± 0.57	97.23 ± 0.42
AM-Softmax (4x steps)	91.93 ± 1.08	95.24 ± 0.60	97.08 ± 0.40
DIAM-Softmax	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

into the problem, we visualize the embeddings via dimensionality reduction. We extract the features of samples as well as the normalized weight vectors of 8 classes[8] in the training dataset and reduce their dimensionality from 512 to 2 using t-SNE [52]. The visualization is shown in Figure 7(a). Noticeably, many weights are shifted from the distribution of the corresponding class even after convergence. Since these weights vectors are the "proxies" to represent the respective class distributions, such a shift could be misleading for updating the features. However, as discussed in Section 4.2, these classifier weights in original AM-Softmax as well as other classification-based methods [51], [50], [49] are updated along with the main feature network using gradients. This process are hence coupled with the settings of the global optimizer. Consequently, this shift become large and harmful on shallow datasets, where there are a large number of classes with only a few samples for most classes. In particular, because SGD updates the network with mini-batches, in a two-shot case, each weight vector will receive attraction signals only twice per epoch. After being multiplied by the learning rate, these sparse attraction signals make little difference to the classifier weights. Thus instead of overfitting, this sparseness of signals from SGD causes the underfitting of the classifier weights in the last fully connected layer, which shift from the feature distribution and lead to the slow convergence.

Based on the above observations, we propose a different optimization method for the weights in classification-based embedding learning loss functions. The main idea is to update the weights based on sample features to avoid underfitting of the classifier weights and accelerate the convergence. This idea of weight imprinting has been studied in the literature [45] [14], but they only imprint the weights at the beginning of fine-tuning. Inspired by the center loss [53], we propose a dynamic weight imprinting (**DWI**) strategy for updating the weights:

$$\mathbf{w}_\mathrm{j} = \frac{\mathbf{w}_\mathrm{j}{}^*}{\| \mathbf{w}_\mathrm{j}{}^* \|_2}, \tag{4}$$

where

$$\mathbf{w}_\mathrm{j}{}^* = (1-\alpha)\, \mathbf{w}_\mathrm{j} + \alpha\, \mathbf{w}_\mathrm{j}^{\mathrm{batch}} \tag{5}$$

Here $\mathbf{w}_\mathrm{j}^{\mathrm{batch}}$ is a target weight vector that is computed based on current mini-batch. Notice that we only update the weights of classes whose samples are present in the current mini-batch and we store $\mathbf{w}_\mathrm{j}$ rather than $\mathbf{w}_\mathrm{j}{}^*$ as variables. We consider three candidates for $\mathbf{w}_\mathrm{j}^{\mathrm{batch}}$ in our ID-selfie problem: (1) the feature of ID image, (2) the feature of selfie image and (3) the mean feature of ID and selfie images. The hyper-parameter α is the update rate. We are using this α here to consider a broader case where the weights are softly updated. In fact, as shown in Section 5.2, $\alpha = 1$ actually leads to the best performance, in which case the update formula can be simply written as:

$$\mathbf{w}_\mathrm{j} = \frac{\mathbf{w}_\mathrm{j}^{\mathrm{batch}}}{\| \mathbf{w}_\mathrm{j}^{\mathrm{batch}} \|_2} \tag{6}$$

Intuitively, DWI helps to accelerate the updating of weight vectors by utilizing sampled features and is invariant to the parameter settings of optimizers. Compared with gradient-based optimization, it only updates the weights based on genuine samples and does not consider repulsion from other classes. This may raise doubts on whether it could optimize the loss function in Equation (1). However, as shown in Figure 6 and Figure 7, empirically we found DWI is not only able to optimize the loss function, but it also helps the loss converge much faster by reducing the shift of weights. Furthermore, our cross-validation results on the Private ID-selfie dataset show that DWI is superior to SGD in terms of accuracy even when we train SGD for twice as long, until complete convergence. See Table 1.

Notice that DWI is not specially designed for AM-Softmax, but can be applied to other classification-based embedding learning loss functions as well, such as L2-Softmax [49]. Although it is mainly an optimization method and it does not change the formula of the loss function, from another perspective, DWI also results in a new loss function, since different choices of classifier weights essentially poses different learning objectives. Therefore, we name the method used in this paper, which combines DWI and AM-Softmax, as a new loss function, called Dynamically Imprinted AM-Softmax (**DIAM-Softmax**).

It is important to note here that DWI does not introduce any significant computational burden and the training speed is almost the same as before. In addition, since DWI only updates the weights of classes that are present in the mini-batch, it is naturally compatible with extremely wide datasets where the weight matrix of all classes is too large to be loaded and only a subset of weights can be sampled for training, as in [14]. However, because of the data limitations, we do not further explore this idea here.

4.4 Domain Specific Modeling

The ID-selfie matching problem can be regarded as an instance of heterogeneous face recognition (HFR) [35], since the face images come from two different sources. Thus, it is reasonable to expect that HFR methods could help with our problem. A common approach in HFR is to utilize two separate domain-specific models to map images from different sources into a unified feature space. Therefore, we use a pair of *sibling networks* for ID images and selfie images, respectively, which share the same architecture but could have different parameters. Both of their features are transferred from the base model, i.e. they have the same initialization. Although this increases the model size, the inference speed will remain unchanged as each image is only fed into one of the sibling networks. The use of sibling networks allows domain-specific modeling, but more parameters could also lead to a higher risk of overfitting. Therefore, different from our prior work [13], we propose to constrain the high-level parameters of the sibling networks to be shared to avoid overfitting. In particular, we use a pair of Face-ResNet models with only the bottleneck layer shared.

8. Seven classes are randomly selected and one class is chosen to have large number of samples. Similar to most classes, the randomly selected seven classes have only two images (one ID and one selfie).

4.5 Data Sampling

When we apply classification-based loss functions to general face recognition, the mini-batches for training are usually constructed by sampling images uniformly at random. However, because our data is acquired from two different domains, such an image-level uniform sampling may not be the optimal choice. There is usually only one ID image per class while there could be many more selfies, so sampling images uniformly could lead to a bias towards selfies and hence an insufficient modeling of the ID domain. Therefore, we propose to use a different sampling strategy to address the domain imbalance problem. In each iteration, $B/2$ classes are chosen uniformly at random, where B is the batch size, and a random ID-selfie pair is sampled from each class to construct the mini-batch. Empirical results in Section 5.3 show that such a balanced sampling leads to a better performance compared with image-level uniform sampling.

5 Experiments

5.1 Experimental Settings

We conduct all the experiments using Tensorflow library[9]. When training the base model with original AM-Softmax on MS-Celeb-1M, we use a batch size of 256 and keep training for 280K steps. We start with a learning rate of 0.1, which is decreased to 0.01, 0.001 after 160K and 240K steps, respectively. When fine-tuning on the Private ID-selfie dataset, we use a batch size of 248 and train the sibling networks for $4,000$ steps. We start with a lower learning rate of 0.01 and decrease the learning rate to 0.001 after $3,200$ steps. For both training stages, the feature networks are optimized by a Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 0.0005. All the images are aligned via similarity transformation based on landmarks detected by MTCNN [54] and are resized to 96×112. We set margin parameters m as 5.0 in both stages. All the training and testing experiments are run on a single Nvidia Geforce GTX 1080Ti GPU with 11GB memory. The inference speed of our model on this GPU is 3ms per image.

By utilizing the MS-Celeb-1M dataset and the AM-Softmax loss function in Equation (1), our base model achieves 99.67% accuracy on the standard verification protocol of LFW and a Verification Rate (VR) of 99.60% at False Accept Rate (FAR) of 0.1% on the BLUFR [55] protocol.

Similar to the protocol of LFW [8], we define two views of the Private ID-Selfie dataset for the following experiments. In the development view, we tune the hyper-parameters including learning schedule, optimizer, m and α using 80% random identities for training and 20% for validation. In the evaluation view, the dataset is equally split into 5 partitions for cross-validation. In each fold, one split is used for testing while the remaining are used for training. In particular, $42,873$ and $10,718$ identities are used for training and testing, respectively, in each fold. All the following analysis experiments are conducted on the evaluation view. In addition, we use the whole Public IvS dataset for cross-dataset evaluation. Cosine similarity is used as comparison score for all experiments.

5.2 Dynamic Weight Imprinting

Here we compare the accuracy of cross-validation using different update rates α and different choices of update vector $\mathbf{w}_j^{\text{batch}}$. We note that the purpose of this section is to show how different strategies of weight imprinting would affect the performance of DIAM-Softmax and the results are consistent with our observations in the development view.

9. https://www.tensorflow.org

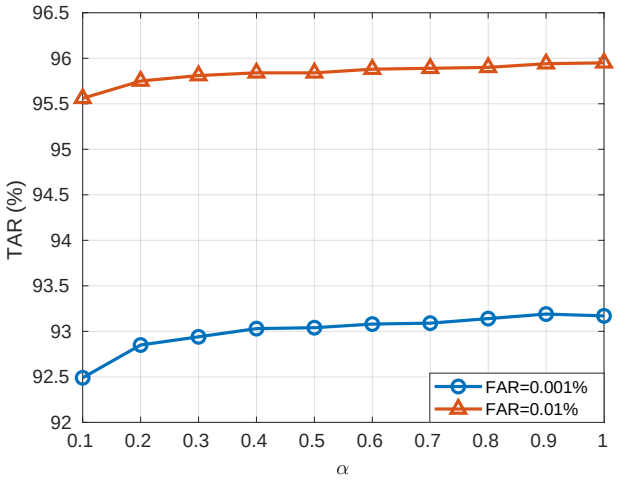


Fig. 8: The mean performance of different values of hyper-parameter α on five folds on the Private ID-selfie dataset.

TABLE 2: The mean (and s.d.) of performance for different choices of $\mathbf{w}_j^{\text{batch}}$ based on 5-fold cross-validation on the Private ID-selfie dataset.

$\mathbf{w}_j^{\text{batch}}$	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
ID features	87.35 ± 1.13	92.87 ± 0.97	96.17 ± 0.51
Selfie features	88.01 ± 1.51	92.96 ± 1.02	96.04 ± 0.68
Average	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

Figure 8 shows how average generalization performance changes along with α. Here, we build the mini-batches with random ID-selfie pairs from different classes. Then, $\mathbf{w}_j^{\text{batch}}$ is chosen as the average feature of the ID and selfie sample. From the figure, it is clear that a larger α always leads to better performance and the accuracy peaks when $\alpha = 1$, where we directly replace the weights with $\mathbf{w}_j^{\text{batch}}$ as in Equation (6). This is not surprising because most classes only have two samples, and thus there is actually no need to update the weights softly since $\alpha = 1$ always leads to the most accurate estimation of the class distribution. A smaller α might be preferred in the case of a deeper dataset.

The results of three different choices of $\mathbf{w}_j^{\text{batch}}$ are shown in Table 2. Using either ID features or selfie features alone leads to a lower performance compared to the averaged feature. This is different from the results of Zhu et al. [14], who found that initializing the classifier weights as ID features leads to the best performance. Such differences may come from different strategies of updating classifier weights since we are updating them dynamically instead of keeping them fixed from the start of training. As most classes have only two images, one from each domain, updating the weights using only one of them causes a biased loss on these images and hence discourages the network from learning better representations for one domain.

5.3 Data Sampling

Since our data can be categorized in two ways: identity and source, it raises the question of how we should sample the

TABLE 3: The mean (and s.d.) of performance of different sampling methods for building mini-batches based on 5-fold cross-validation on the Private ID-selfie dataset. In "Random ID-selfie pairs", a random pair of ID and selfie images are sampled for each class (selected randomly). In "Random pairs", the sampled pairs may come from the same source.

Sampling	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
Random images	92.77 ± 1.05	95.71 ± 0.67	97.41 ± 0.48
Random pairs	92.66 ± 0.93	95.63 ± 0.70	97.27 ± 0.43
Random ID-selfie pairs	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

images for mini-batches during training. Here, we compare three different sampling methods: (1) image-wise random sampling, (2) random pairs from different classes, and (3) random ID-selfie pairs from different classes (proposed). Method (1) is commonly used for training classification loss functions, while method (2) is commonly used by metric learning methods because they require genuine pairs within the mini-batch for training. For (1) and (2), the classifier weights are updated with the average feature of the samples in each class. The corresponding results are shown in Table 2. As one can see, random image sampling works slightly better than random pair sampling, which is consistent with the results on general face recognition. This is because random pair leads to different sampling chances for images in different classes, and the model will be biased towards samples in small classes, which are sampled more frequently. However, in spite of this problem, random ID-selfie pair sampling still work slightly better than random image sampling, which shows that a balanced parameter learning of the two domains is important in our problem. These results imply that one should investigate how to further improve the performance by simultaneously solving the class imbalance problem (or long-tail problem) and domain imbalance problem.

5.4 Parameter Sharing

To evaluate the effect of shared parameters vs. domain-specific parameters, we constrain a subset of the parameters in the sibling networks to be shared between ID and selfie domains and compare the performances. Here, we consider both the case of shared low-level parameters and the case of shared high-level parameters. In particular, we compare the cases where modules "Conv1", "Conv1-3" and "Conv1-5" in the network are shared for learning low-level parameters. Then, we repeat the experiments by sharing the modules "FC" and "Conv5 + FC" and "Conv4-5 + FC" for high-level parameters. Here "Conv i-j" means the i^{th} to j^{th} convolutional modules and "FC" means the fully connected layer for feature extraction. The results are shown in Table 4.

From Table 4 we can see that the performance does not vary a lot with the parameter sharing. This is partially because our base model learns highly transferable features, whose parameters already provides a good initialization for all these modules. In particular, sharing low-level parameters does not have a clear impact on the performance, but constraining a shared bottleneck does lead to a slight improvement in both accuracy and standard deviation at all false accept rates. The performance decreases when we constrain more high-level parameters to be shared. Furthermore, sharing all parameters leads to worse performance when compared with others. We can conclude, indeed, that there exists a small difference between ID photo and selfie photo domains and learning domain-specific parameters is helpful for recognition of photos across the two domains.

TABLE 4: The mean (and s.d.) of performance of constraining different modules of the sibling networks to be shared. "All" indicates a single model for both domains while "None" means that the parameters for the two domains are completely independent.

Shared Modules	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
None	93.07 ± 0.91	95.86 ± 0.56	97.45 ± 0.39
Conv 1	93.08 ± 0.95	95.86 ± 0.58	97.47 ± 0.40
Conv 1-3	93.13 ± 0.88	95.84 ± 0.57	97.46 ± 0.42
Conv 1-4	93.11 ± 0.85	95.85 ± 0.57	97.43 ± 0.41
FC	$\mathbf{93.16 \pm 0.85}$	95.95 ± 0.54	$\mathbf{97.51 \pm 0.40}$
Conv 5 + FC	93.14 ± 0.89	95.96 ± 0.55	97.48 ± 0.43
Conv 4-5 + FC	93.10 ± 0.85	$\mathbf{95.97 \pm 0.57}$	97.46 ± 0.42
All	92.91 ± 0.92	95.81 ± 0.63	97.40 ± 0.43

TABLE 5: The mean (and s.d.) of performance of static weight imprinting and dynamic weight imprinting based on 5-fold cross-validation on the Private ID-selfie dataset. "Static-fixed" updates all the weights at the beginning of fine-tuning. "Static-periodic" updates all the weights every two epochs. For the proposed "DWI", all the weights are randomly initialized.

Weight Update	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
Static - fixed	90.97 ± 1.01	94.76 ± 0.64	96.91 ± 0.46
Static - periodic	92.95 ± 0.85	95.88 ± 0.50	97.43 ± 0.39
DWI	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

5.5 Comparison with Static Weight Imprinting

In [14], Zhu et al. used ID features as fixed classifier weights to run fine-grained training and showed performance improvement from the original representation. We regard such a method as static weight imprinting. The advantage of static weight imprinting is that one can extract the features simultaneously from all classes to update the classifier weights. However, they not only result in extra computational cost but also fail to capture the global distribution during the training. We compare our dynamic weight imprinting method with static imprinting methods. In particular, we consider two cases of static imprinting: (1) updating weights only at the beginning of fine-tuning and keeping them fixed during training and (2) updating weights every two epochs. For static methods, we extract the features of a random ID-selfie pair from every class and use their average vector for updating the weights. The results are shown in Table 5. It can be noted that periodic updating outperforms fixed weights, since fixed weights fail to keep up with the feature distribution. Better performance should be expected if we update the static weights more frequently. However, it is important to note that periodic updating also introduces additional computational cost as we need to extract tens of thousands of features every time we update the weights. In comparison, DWI has almost zero extra computational cost, yet it leads to even better performance than periodic updating, indicating that the weights under the proposed DWI are able to keep up-to-date and capture the global distribution accurately.

5.6 Comparison with Different Loss Functions

In this section, we evaluate the effect of different loss functions for fine-tuning on the Private ID-selfie dataset. We do not delve

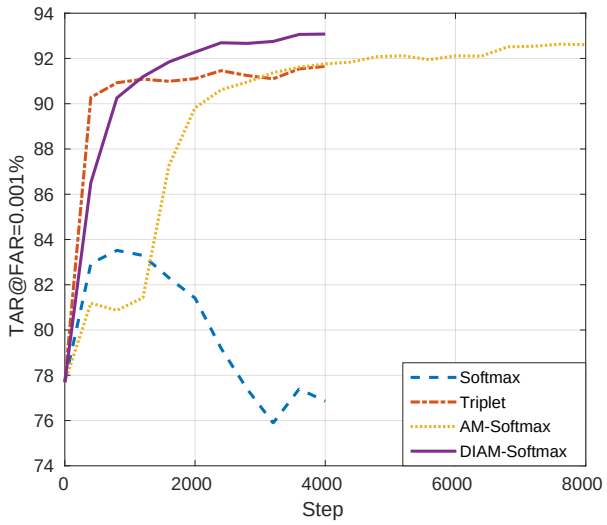


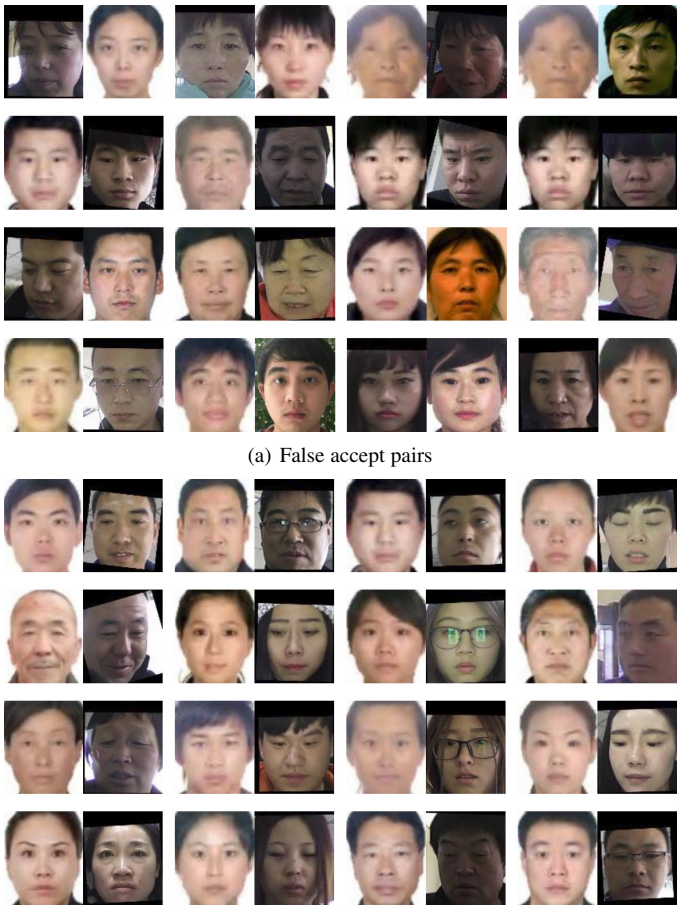
Fig. 9: Generalization performance of different loss functions during training. The x-axis is the number of training steps and the y-axis is the average TAR@FAR=0.001% of five folds at the corresponding step.

TABLE 6: The mean (and s.d.) of performance of different loss functions based on 5-fold fold cross-validation on the Private ID-selfie dataset. "N/C" means not converged. The proposed method is shown in italic style in the last row of the table.

Method	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
Base Model	77.69 ± 2.02	85.95 ± 1.67	92.43 ± 0.98
Softmax	83.51 ± 1.76	90.14 ± 1.44	94.53 ± 0.81
A-Softmax [21]	71.05 ± 2.57	80.26 ± 0.24	88.36 ± 1.76
AM-Softmax [23]	92.53 ± 1.09	95.57 ± 0.57	97.23 ± 0.42
Contrastive [42]	91.13 ± 1.65	95.05 ± 0.77	97.18 ± 0.47
Triplet [19]	91.68 ± 1.21	95.42 ± 0.70	97.26 ± 0.45
MPS [42]	91.79 ± 1.16	95.43 ± 0.65	97.27 ± 0.44
DIAM-Softmax	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

into the choice of loss function for training the base model since it is not relevant to the main topic of this paper. We compare the proposed DIAM-Softmax with three classification-based embedding learning loss functions: Softmax, A-Softmax [21] and AM-Softmax [23] and three other metric learning loss functions: contrastive loss [42], triplet loss [19] and the MPS loss proposed in our prior work [13]. The classification-based loss functions are able to capture global information and thus achieve state-of-the-art performances on general face recognition problems [23] while the metric learning loss functions are shown to be effective on very large datasets [19]. To ensure a fair comparison, we implement all the loss functions in Tensorflow and keep the experimental settings the same except that AM-Softmax is trained for twice as long. The results are shown in Table 6.

From Table 6, one can see that our base model already achieves quite high performance (TAR of $92.43\% \pm 0.98$ at FAR of 0.1%) on the target dataset without any fine-tuning, indicating that the features learned on general face datasets are highly transferable, but it is still significantly lower than its performance on general face datasets such as LFW due to the discrepancy between the characteristics of face images in these two tasks. Clear improvement can be observed after fine-tuning with most of the loss functions. To gain more insight, we plot the TAR-step curves (the x-axis is the number of training steps and the y-axis is the mean TAR@FAR=0.001% of five folds at that step) in Figure 9. We only pick the representative loss functions for clarity of the plot. From Figure 9, we can see that Softmax overfits heavily just after two epochs[10], while the metric learning loss (triplet loss) is more stable and quick to converge. Although AM-Softmax performs better than metric learning methods, it converges so slowly that we have to train it for twice as many steps. Notice that this result is not contradictory with our prior work [13], where we found AM-Softmax perform poorly on fine-tuning, because we only allowed an equally limited number of training steps in [13]. A-Softmax, because of the multiplicative margin, is unstable and does not converge with its default parameters [21]. Although it converges after we change its margin parameter to 3, it ends with a lower performance than the base model. In comparison with the slow convergence of AM-Softmax, with the proposed weight updating method, the "DIAM-Softmax" not only converges faster, it is also robust against overfitting and generalizes much better than all competitors.



(a) False accept pairs

(b) False reject pairs

Fig. 10: Examples of falsely classified images by our model on the Private ID-selfie dataset at FAR = 0.001%.

5.7 Comparison with existing methods

In this subsection, we evaluate the performance of other face matchers on the Private ID-selfie dataset to compare with our method. To the best of our knowledge, there are no public face matchers in the domain of ID-selfie matching. Although Zhu et

10. Each epoch is about 300 steps

TABLE 7: The mean (and s.d.) of performance of different matchers on the private ID-selfie dataset. The "base model" is only trained on MS-Celeb-1M. The model *DocFace+* has been fine-tuned on training splits. Our models are shown in italic style. The two COTS and two public CNN face matchers are not retrained on our dataset.

Method	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
COTS-1	58.62 ± 2.30	68.03 ± 2.32	78.48 ± 1.99
COTS-2	91.53 ± 1.96	94.41 ± 1.84	96.50 ± 1.78
CenterFace [53]	27.37 ± 1.27	41.38 ± 1.43	59.29 ± 1.42
SphereFace [21]	7.96 ± 0.68	21.15 ± 1.63	50.76 ± 1.55
InsightFace [56]	81.69 ± 1.73	88.78 ± 1.30	94.08 ± 0.78
Base model	77.69 ± 2.02	85.95 ± 1.67	92.43 ± 0.98
DocFace+	$\mathbf{93.16 \pm 0.85}$	$\mathbf{95.95 \pm 0.54}$	$\mathbf{97.51 \pm 0.40}$

TABLE 8: Evaluation results on Public-IvS dataset. The model *DocFace+* has been fine-tuned on the entire Private ID-selfie dataset and no training is involved on Public-IvS dataset. The performance of [14] is reported in their paper. Our models are shown in italic style.

Method	True Accept Rate (%)		
	FAR=0.001%	FAR=0.01%	FAR=0.1%
COTS-1	83.78	89.92	92.90
COTS-2	94.74	97.03	97.88
CenterFace [53]	35.97	53.30	69.18
SphereFace [21]	53.21	69.25	83.11
InsightFace [56]	**96.48**	**98.30**	**99.06**
Zhu et al. [14]	93.62	97.21	98.83
Base model	95.00	97.71	98.80
DocFace+	91.88	96.48	98.40

al. developed a system on 2.5M ID-selfie training pairs, both their system and training data are not in the public domain and therefore we cannot compare their system with the proposed method on our dataset. Therefore, we compare our approach with state-of-the-art general face matchers to evaluate the efficacy of our system on the problem of ID-selfie matching. To make sure our experiments are comprehensive enough, we compare our method not only with two Commercial-Off-The-Shelf (COTS) face matchers, but also three state-of-the-art open-source CNN face matchers, namely CenterFace[11] [53], SphereFace[12] [21] and InsightFace[13](ArcFace) [56]. During the five-fold cross-validation, because these general face matchers cannot be retrained, only the test split is used. The results are shown in Table 7. Performances of the two open-source CNN matchers, CenterFace and SphereFace, are below par on this dataset, much worse than their results on general face datasets [53] [21]. Although our base model performs better, it still suffers from a large drop in performance compared to its performance on general face datasets. It can be concluded that general CNN face matchers cannot be directly applied to the ID-selfie problem because the characteristics of ID-selfie images are different than those of general face datasets and a domain-specific modeling is imperative. A commercial state-of-the-art face recognition system, COTS-2, performs closer to our fine-tuned model. However, since the face dataset used to train COTS-2 is proprietary, it is difficult to conclude whether a general commercial face matcher can work well on this problem. In fact, from Table 7, another commercial face matcher, COTS-1, performs much worse on this dataset.

5.8 Evaluation on Public-IvS

In [14], Zhu et al. released a *simulated* ID-selfie dataset for open evaluation. The details and example images of this dataset were given in Section 3.3. Here, we test our system as well as the previous public matchers on this dataset for comparison. Among all the $5,503$ photos, we were able to successfully align $5,500$ images with MTCNN. Assuming that the subjects are cooperative and no failure-to-enroll would happen in real applications, we only test here on the aligned images. The DocFace+ model is trained on the entire Private ID-selfie dataset. Hence, no cross-validation is needed here. The results are shown in Table 8. Contrary to our expectations, the general face matchers perform much better on this dataset. Furthermore, our base model even outperforms the fine-tuned one. These results can likely be attributed to the high quality of the simulated ID images in this dataset. Consequently, this dataset is similar to other general face datasets, such as LFW [8]. See Figure 3 for a comparison of the images in the Private ID-selfie dataset which is available to us and those in the Public-IvS dataset. The example failure cases of our system (DocFace+) on this dataset, shown in Figure 11, also supports this conjecture. As one can observe, many ID images in these failure pairs are more like a normal frontal face photo than a real ID image, which could favor general face matchers. Another possible case is that both the private ID-selfie dataset and the Public IvS dataset represent a different subset of the overall distribution of ID-selfie images, thus leading to different results. Notice that although Zhu et al. [14] found that the results on this dataset are coherent with real ID-selfie datasets, they did not compare them with general face matchers and hence they did not observe the competitiveness of general matchers on this dataset. However, we can still conclude that our DocFace+ system is robust and it performs well for both the private ID-selfie dataset and Public IvS dataset.

11. https://github.com/ydwen/caffe-face
12. https://github.com/wy1iu/sphereface
13. https://github.com/deepinsight/insightface

6 Conclusions

In this paper, we propose a new face recognition system, named DocFace+, for matching ID document photos to selfies. The transfer learning technique is used, where a base model for unconstrained face recognition is fine-tuned on a private ID-selfie dataset. A pair of sibling networks with shared high-level modules are used to model domain-specific parameters. Based on our observation of the weight-shift problem of classification-based embedding learning loss functions on shallow datasets, we propose an alternative optimization method, called dynamic weight imprinting (DWI) and a variant of AM-Softmax, DIAM-Softmax. Experiments show that the proposed method not only helps the loss converge much faster but also leads to better generalization performance. A comparison with static weight imprinting methods confirms that DWI is capable of capturing the global distribution of embeddings accurately. Different sampling methods are studied for mini-batch construction and we find that a balanced sampling between the two domains is most helpful for learning generalizable features. We compare the proposed system with existing general face recognition systems on our private dataset and see a significant improvement with our system, indicating the necessity of domain-specific modeling of ID-selfie data. Finally, we compare the performance of different matchers on the Public-IvS dataset and find that although this dataset is similar to a general face dataset, our system still generalizes well.

(a) False accept pairs

(b) False reject pairs

Fig. 11: Examples of falsely classified images by our model on the Public ID-selfie dataset at FAR = 0.001%.

ACKNOWLEDGEMENT

This work is partially supported by Visa, Inc. We also thank ZKTeco for providing the ID-Selfie datasets.

REFERENCES

[1] D. White, R. I. Kemp, R. Jenkins, M. Matheson, and A. M. Burton, "Passport officers errors in face matching," *PloS one*, 2014.
[2] Wikipedia, "Australia smartgate," https://en.wikipedia.org/wiki/SmartGate, 2018.
[3] Wikipedia, "epassport gates," https://en.wikipedia.org/wiki/EPassport_gates, 2018.
[4] U.S. Customs and Border Protection, "Automated passport control (apc)," https://www.cbp.gov/travel/us-citizens/apc, 2018.
[5] Xinjiang Heng An Perimeter Security Equipment Co., "What is id-person matching?" http://www.xjhazj.com/xjhazj/vip_doc/8380983.html, 2018.
[6] Jumio, "Netverify id verification," https://www.jumio.com/trusted-identity/netverify, 2018.
[7] Mitek, "Mitek id verification," https://www.miteksystems.com/mobile-verify, 2018.
[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
[9] V. Starovoitov, D. Samal, and B. Sankur, "Matching of faces in camera images and document photographs," in *ICASSP*, 2000.
[10] T. Bourlai, A. Ross, and A. Jain, "On matching digital face images against scanned passport photos," in *IEEE Int. Conf. Biometrics, Identity and Security (BIDS)*, 2009.
[11] T. Bourlai, A. Ross, and A. K. Jain, "Restoring degraded face images: A case study in matching faxed, printed, and scanned photos," *IEEE Trans. on TIFS*, 2011.
[12] V. Starovoitov, D. Samal, and D. Briliuk, "Three approaches for face recognition," in *International Conference on Pattern Recognition and Image Analysis*, 2002.
[13] Y. Shi and A. K. Jain, "Docface: Matching id document photos to selfies," in *BTAS*, 2018.
[14] X. Zhu, H. Liu, Z. Lei, H. Shi, F. Yang, D. Yi, and S. Z. Li, "Large-scale bisample learning on id vs. spot face recognition," *arXiv:1806.03018*, 2018.
[15] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *IEEE Conf. on Automatic Face & Gesture Recognition*, 2018.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
[17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014.
[18] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *NIPS*, 2014.
[19] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.
[20] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *ECCV*, 2016.
[21] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017.
[22] A. Hasnat, J. Bohné, J. Milgram, S. Gentric, and L. Chen, "Deepvisage: Making face recognition simple yet with powerful generalization skills," *arXiv:1703.08388*, 2017.
[23] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *arXiv:1801.05599*, 2018.
[24] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, 2016.
[25] W. Liu, Y.-M. Zhang, X. Li, Z. Yu, B. Dai, T. Zhao, and L. Song, "Deep hyperspherical learning," in *NIPS*, 2017.
[26] S. Z. Li and A. Jain, *Encyclopedia of biometrics*. Springer Publishing Company, Incorporated, 2015.
[27] S. Z. Li, R. Chu, S. Liao, and L. Zhang, "Illumination invariant face recognition using near-infrared images," *IEEE Trans. on PAMI*, 2007.
[28] J. Choi, S. Hu, S. S. Young, and L. S. Davis, "Thermal to visible face recognition," in *SPIE DSS-DS107: Biometric Technology for Human Identification IX*, 2012.
[29] X. Tang and X. Wang, "Face photo recognition using sketch," in *ICIP*, 2002.
[30] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Trans. on PAMI*, 2008.
[31] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," in *CVPR*, 2005.
[32] X. Gao, J. Zhong, J. Li, and C. Tian, "Face sketch synthesis algorithm based on e-hmm and selective ensemble," *IEEE Trans. on Circuits and Systems for Video Technology*.
[33] S. Liao, D. Yi, Z. Lei, R. Qin, and S. Z. Li, "Heterogeneous face recognition from local structures of normalized appearance," in *ICB*, 2009.
[34] B. Klare and A. K. Jain, "Heterogeneous face recognition: Matching nir to visible light images," in *ICPR*. IEEE, 2010.
[35] B. F. Klare and A. K. Jain, "Heterogeneous face recognition using kernel prototype similarities," *IEEE Trans. on PAMI*, 2013.
[36] X. Liu, L. Song, X. Wu, and T. Tan, "Transferring deep representation for nir-vis heterogeneous face recognition," in *ICB*, 2016.
[37] R. He, X. Wu, Z. Sun, and T. Tan, "Learning invariant deep representation for nir-vis face recognition." in *AAAI*, 2017.
[38] X. Wu, L. Song, R. He, and T. Tan, "Coupled deep learning for heterogeneous face recognition," *arXiv: 1704.02450*, 2017.
[39] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML, Workshop on Deep Learning*, 2015.
[40] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NIPS*, 2016.
[41] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.
[42] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.
[43] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.
[44] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *NIPS*, 2016.
[45] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018.

[46] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," 2017.
[47] 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.
[48] 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," *arXiv:1801.09414*, 2018.
[49] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv:1703.09507*, 2017.
[50] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: l_2 hypersphere embedding for face verification," *arXiv:1704.06369*, 2017.
[51] M. Hasnat, J. Bohné, J. Milgram, S. Gentric, L. Chen *et al.*, "von mises-fisher mixture model-based deep learning: Application to face verification," *arXiv:1706.04264*, 2017.
[52] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, 2008.
[53] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016.
[54] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, 2016.
[55] S. Liao, Z. Lei, D. Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," in *IJCB*, 2014.
[56] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv:1801.07698*, 2018.

Yichun Shi received his B.S degree in the Department of Computer Science and Engineering at Shanghai Jiao Tong University in 2016. He is now working towards the Ph.D. degree in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and computer vision.

Anil K. Jain is a University distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. He served as the editor-in-chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence (1991-1994) and as a member of the United States Defense Science Board. He was elected to the National Academy of Engineering, Indian National Academy of Engineering and The World Academy of Sciences (TWAS) for advancement of science in developing countries.