Exploring the Limits of Hard Example Mining for ID Document to Selfie Matching

Zichang Tan, *Member, IEEE*, Ajian Liu, Jun Wan, *Senior Member, IEEE*, Zhen Lei, *Senior Member, IEEE*, and Guodong Guo, *Senior Member, IEEE*

***Abstract*—Most face verification systems verify a person's identity by comparing the ID document with the live face (also called spot face). More specifically, the spot face can be regarded as the probe image and the face in ID document can be regard as the reference image. The identity verification is then conducted by calculating the similarity of the two images. This problem is called ID vs. Spot (IvS) face recognition. IvS face recognition is different from general face recognition, where the IvS face datasets usually contain a very large number of identities (up to millions and more) with only two images per identity. We adopt a metric learning way rather than a classification-based way to train the network on IvS datasets, aiming to avoid tremendous pressure on GPU resource. In this work, we improve the performance mainly from two aspects, namely extending the selecting space to select very hard samples and increasing the diversity of selected samples. For the former, we propose a Super Batch (S-Batch) by aggregating many traditional batches together, in which each anchor sample can select the hard sample pairs from a very large batch. Moreover, we employ a Cross-Batch Hard Example Mining (CBHEM) to select hard samples from not only current batch but also historical batches. For the latter, we propose a Various Batch Sizes (VBS) in our S-Batch, which selects hard samples of different batch scales for training.Extensive experiments demonstrate the effectiveness of our method on IvS face recognition.**

***Index Terms*—Deep learning, hard example mining, IvS face recognition.**

I. Introduction

IDENTITY verification, which aims to verify a person's identity whether matches the one that is supposed to be, has wide range of applications in our daily lives, e.g., access control, international border control and financial security. In many real world applications, the identity verification is usually achieved by matching ID document photo to the holder's live face photo (called as selfie[1] or spot face) as shown in Fig 1. For example, the face authentication system in Chinese railway station checks the identity by matching ID photos and live faces. Similar authentication systems also can be found in ePassport gates, ID card gates and so on. Following the work [2], the live face photo is also called spot face, and this kind of verification is called as ID vs. Spot (IvS) face recognition [2].

Owing to the rapid development of deep learning [3], [4], recent years has witnessed the great success of face recognition in network constructing [5], [6], algorithm designing [7], [8], [9], [10], face data collection [11], [12], [13]. However, the IvS face recognition is largely different from the general face recognition and still encounters enormous challenges. Generally, there are mainly three differences between the IvS face recognition and the general face recognition:

- *Large Data Bias:* The images for general face recognition are usually collected from the scene in the wild, showing a good quality with high resolution and clear lighting. For IvS dataset, ID faces are captured from constrained environments with small variations, while Spot faces are captured by verification systems in unconstrained environments, and usually show large variations in pose, lighting, occlusions and so on. Besides, ID faces have low resolution due to image compression in ID document.
- *Heterogeneity:* The pair of images (i.e., one ID face and one Spot face) are captured from different environments and show different characteristics. Such heterogeneity further increases the difficulty of IvS face recognition.
- *Data Structures:* In general face recognition, face datasets usually contain limited identities (less than 200,000) and

[1]Following the definition in the work [1], "selfies" refer to any self-captured live face photos, including those from mobile phones and kiosks.

Manuscript received 18 October 2021; revised 4 March 2022; accepted 20 June 2022. Date of publication 26 July 2022; date of current version 5 December 2022. This work was supported in part by the National Key Research and Development Plan under Grant 2021YFE0205700; in part by the External Cooperation Key Project of Chinese Academy Sciences under Grant 173211KYSB20200002; in part by the Chinese National Natural Science Foundation Projects under Grant 61876179, Grant 61961160704, Grant 61876178, Grant 62106264, Grant 62176256, and Grant 61976229; in part by the Science and Technology Development Fund of Macau Project under Grant 0070/2020/AMJ; in part by the Open Research Projects of Zhejiang Lab under Grant 2021KH0AB07; and in part by InnoHK Program. This article was recommended for publication by Associate Editor N. Evans upon evaluation of the reviewers' comments. *(Corresponding author: Guodong Guo.)*

Zichang Tan and Guodong Guo are with the Institute of Deep Learning, Baidu Research, Beijing 100000, China, and also with the National Engineering Laboratory for Deep Learning Technology and Application, Beijing 100000, China (e-mail: tanzichang@baidu.com; guodong.guo@mail.wvu.edu).

Ajian Liu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: ajianliu92@gmail.com).

Jun Wan is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: jun.wan@ia.ac.cn).

Zhen Lei is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China, and also with the Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hong Kong, China (e-mail: zlei@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TBIOM.2022.3193865

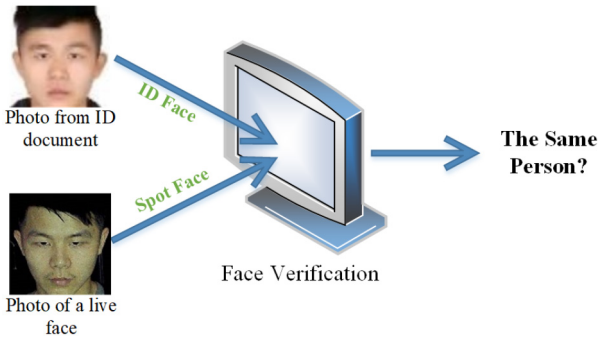


Fig. 1. A simple illustration of IvS face recognition. The system takes two photos as the input, with one photo from ID document and the other from a live face. Then, the system judges whether the two input images come from a same person or not.

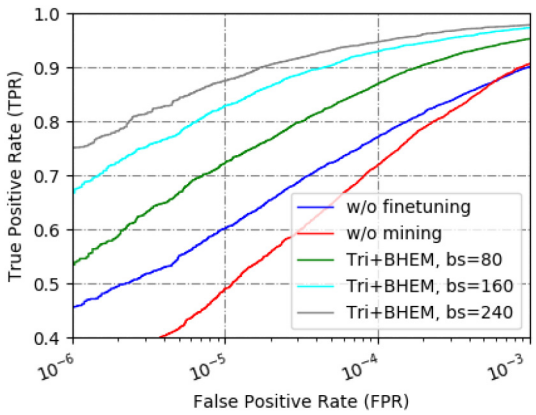


Fig. 2. 'Comparisons of using different training settings. 'w/o finetuning' means the model only trained on MS-Celeb-1M dataset. Besides this, all other settings are first pretrained on MS-Celeb-1M dataset and then finetuned on the IvS dataset. 'w/o mining' means training without using hard example mining in Triplet loss. 'Tri + BHEM' means training with Triplet loss and Batch Hard Example Mining (BHEM), and 'bs' indicates the batch size.

each identity contains adequate images. For IvS face recognition, the face data is usually collected by practical system (like railway station), where **a very large number of identities (up to millions and more) are available but only two images (an ID face and a live face) per identity are accessible.**

Due to the first two differences between the IvS and general face recognition, namely the data bias and heterogeneity, directly applying the face model of general face recognition (i.e., the model is trained on general face datasets, like MS-Celeb-1M) to the IvS face recognition results into the poor performance, which is reflected from Fig. 2 (see the setting of 'w/o finetuning'). Therefore, a new training scheme is urgently needed for IvS face recognition. However, due to the distinctive data structure of the IvS dataset, training a face model on the IvS dataset is not an easy thing. On one hand, current methods for general face recognition are mainly based on classification, which are inapplicable for the IvS dataset of containing massive where a huge number of parameters in the classification layer are created. For example, for a softmax layer with 2 million prototypes, about 8GB GPU memory is needed to only save their values and derivatives. Besides, at least tens of GPU memory are needed to conduct matrix operation. Certainly, such a large demand on GPU memory is infeasible for most computing devices. On the other hand, each identity only has insufficient images (one ID face and One spot face), where the intra-variations will be hardly learned and the derived feature space would not be discriminative enough in the test stage as clarified in [2]. For facing massive identities, the metric learning is a feasible scheme where the classification layer can be removed and numerous parameters are avoided. For each identity containing insufficient images (in particular two images for each identity), it is rarely involved in current studies.

In this paper, our goal is to seek an effective training scheme for the IvS face recognition, in which the dataset contains massive identities but each identity contains inadequate images. As we have mentioned above, we train the model following a metric learning manner. Specifically, we follow previous works [14], [15], [16] to adopt the classic Triplet loss [15] as the loss function due to its effectiveness for face recognition. Considering the training difficulty caused by the insufficient images of each identity, we follow previous works [1], [2], [17] to adopt a general-to-specific scheme where the model is first pretrained on a general face dataset like MS-Celeb-1M to obtain a good weight initialization and then adapted to the specific domain of IvS face recognition. One useful training strategy in metric learning is hard example mining [18], [19], e.g., Batch Hard Example Mining (BHEM) [19], in which the hard and useful sample pairs are employed for training while the easy and useless sample pairs are discarded. We also take some experiments with Triplet loss and BHEM as shown in Fig. 2. From experimental results, we find that the hard example mining really helps to capture effective features and employing a larger batch size can achieve the better performance. In BHEM, each sample is paired with a hardest positive and negative samples (with smallest and largest distances, respectively) within a batch to form a hard triplet, and therefore, using a larger batch size indicates harder sample pairs can be selected where more samples in the selecting space can be chosen from. However, BHEM selects hard sample pairs only from the current batch, but the batch size still can not be set too large due to the limited GPU memory size (e.g., only 11 GB for GTX 2080Ti), which limits the selecting scope for each sample.

Inspired by above observations, one reliable way to improve the performance is to extend the selecting space and select very hard samples for training. To achieve this, we formulate two training strategies, namely Super Batch (S-Batch) and Cross-Batch Hard Example Mining (CBHEM) [20], [21]. For S-Batch, it aims to aggregate many traditional batches together to form a larger batch. It consists of two steps. In the first step, the samples of K batches are fed to the network and all features are recorded by a memory bank. On the basis of this, all the samples of adjacent K batches are visible to each sample and the hard samples across K batches can be found out and recorded. In the second step, we calculate the loss according to the above selected hard triplets batch by batch and use the accumulated gradients of all batches to update the network. Note that the loss function for each batch is carefully designed to let it is equivalent to the loss of training with a real large

batch size. For CBHEM, we consider historical features from previous batches (or S-Batches)also can be used as a reference to select hard samples although the network are changing throughout the training process. Specifically, we further use a queue to collect the features of several previous batches, in which each sample can select the hard sample from not only the current batch but also previous batches. Moreover, setting a large batch size will reduce the diversity of selected hard triplets due to the dataset size is limited. Consider an extreme example where we set the batch size as the dataset size, the hardest negative sample for each anchor is often the same image in the training process, where the diversity of select hard triplets is reduced, which prevents the model from receiving effective training. To increase the diversity of select hard triplets, we propose a Various Batch Sizes (VBS) in S-Batch, which selects the hard triplets for each anchor at different batch scales for training. Finally, we integrate the aforementioned components together to train a strong network for IvS face recognition.

The main contributions can be summarized as follows:

- We propose a Super Batch (S-Batch) with a two forward stages. It can virtually increase the batch size to a very large number, with breaking through the limitation of GPU memory.
- We propose a Various Batch Sizes (VBS) to increase the diversity of select hard triplets.
- Extensive evaluations demonstrate the superiority of the proposed method over several benchmark datasets including Private-IvS, Public-IvS and LFW-BLUFR.

II. Related Works

General Face Recognition: In recent years, deep leaning [3], [4], [22], [23], [24], [25] has achieved great successes in face recognition [7], [8], [8], [13], [16], [26], [27], [28], [29], [30], [31]. At the beginning, researchers usually regard face recognition as a classification problem, e.g., DeepFace [26] and DeepID [27]. After that, Sun *et al.* [27] and Schroff *et al.* [15] learns face representations by using a joint identification-verification supervision signal and Triplet loss, respectively. Later, researchers find that loss function is very important to learn better feature representations and lots of loss functions are proposed, e.g., Center loss [7], A-softmax (ShpereFace) [28], SphereFace+ [32], AM-softmax [29] ArcFace [16], rotation consistent margin (RCM) loss [33] , CurricularFace [9] and UniformFace [34]. Specifically, A-softmax, AM-softmax and ArcFace learn better representations by adding angular or cosine margin constraints in the loss function, with leading to the start-of-the-art performance. SphereFace+ [32] is proposed to apply a minimum hyperspherical energy (MHE) to SphereFace, which clearly improves the inter-class feature separability and achieves considerable improvements. UniformFace [34] is proposed to learn deep equidistributed representations, which uniformly spreads the class centers on the manifold and maximizes the minimum average interclass distances for all the classes. Moreover, some works [35], [36] introduce new training paradigms for face recognition. For example, SphereFace-R unifies hyperspherical face recognition by a general principle for a loss function to incorporate large angular margins, and SphereFace2 [36] builds a novel binary classification framework rather than multi-class classification framework for face recognition. There are also some works improve the performance by adopting attention mechanism [37], [38], [39], better network architecture [40], [41] and so on. For example, Wang *et al.* [38] propose a hierarchical pyramid diverse attention (HPDA) network to learn multi-scale diverse local representations automatically and adaptively. Wu *et al.* [41] propose a light CNN framework to learn a compact embedding on the large-scale face data. Some researchers improve face recognition against face aging [42], [43] or pose variations [44], [45]. For example, Hou *et al.* [42] disentangle the face representations into identity-dependent and age-dependent components for age-invariant face recognition. Moreover, some researches are proposed to improve the face recognition on noise data [6], [46], [47] or synthetic data [48]. However, most of those works didn't consider a special case in face recognition, i.e., IvS face recognition, where the training dataset contains massive identities but each identity only contains two image samples. Therefore, most methods proposed for general face recognition are usually not suitable fo IvS face recognition due to limited GPU memory and the difficulties of learning effective representations.

IvS Face Recognition: IvS face recognition is a special case in face recognition. The datasets of IvS face recognition usually contain a very large number of identities and each identity only has two images. Therefore, those classification-based methods in general face recognition (e.g., A-softmax, AM-softmax and ArcFace) may hardly be implemented on a single machine when training with a large number classes (e.g., one million classes), where the classifier will occupy a lot of GPU memory. To avoid posing tremendous pressure on GPU resource, Zhu *et al.* [2] propose a dominant prototype softmax (DP-softmax) to select a small number of dominant classes from CPU to participate into the classification each iteration. DP-softmax is still a classification-based method. However, each identity only contains two images and the learned feature space would not be discriminative enough if classification method is adopted for training. Moreover, Shi and Jain [1], [17] propose a DIAM-Softmax for IvS face recognition with training on a small dataset with only 53,591 identities. However, DIAM-Softmax is also a classification-based loss function, and it is hard to train the network with limited GPU memory when encountering with a large number of identities. IvS face recognition is a very important topic in face recognition due to its many applications in real life. However, the relevant research in this field is very few and lots of challenges need to be solved urgently. Considering this, a Super-Batch and a CBHEM are proposed for training IvS face recognition with massive identities.

Deep Metric Learning: Deep metric learning [15], [20], [49], [50], [51], [52], [53], [54], [55], [56], [57] optimizes the deep feature space by calculating pairwise distances or similarities, e.g., Contrastive loss [49], Triplet loss [15], quadruplet loss [58], histogram loss [51]. For example, Triplet loss [15] conducts the feature learning on a positive pair and a negative pair, and both of them are built on the same anchor sample. Later, researchers propose new losses to more fully

exploit pair-wise relations between samples in a mini-batch, e.g., N-pair loss [52], Lifted Structure loss [50], Ranked List loss [56], Multi-Similarity (MS) loss [55], proxy-based losses [54], [57], [59]. For example, N-pair loss is proposed to associate an anchor sample with a positive sample and multiple negative samples, and Ranked List loss considers all positive and negative samples in a batch. However, considering all samples equally is not optimal for metric learning because some of those sample pairs (e.g., easy pairs) contribute very little to the feature learning. Therefore, Multi-Similarity (MS) loss [55] considers all pairs in a batch with assigning a weight to each sample pair in the learning stage, where contributions of useful pairs would be enhanced and contributions of useless pairs would be suppressed. Further, proxy-based losses are proposed to reduce the training complexity by introducing proxies, where each data point is only associated with proxies. For example, Proxy Anchor Loss [57] associates all data with each proxy by considering their relative hardness determined by data-to-data relations. Besides designing objective functions for deep metric learning, some improved architectures [60], [61] are also proposed to learn better embeddings.

Moreover, the strategy of hard example mining [18], [19], [62] is proposed to only select the most difficult sample pairs for training while dropping easy pairs. Recently, some adaptive sampling strategies like Policy-Adapted Sampling [63] and Smart Mining [18] are proposed to adaptively select the most effective samples for training. Some researchers also propose generation-based methods [64], [65] to create synthetic samples, which act as hard negatives to improve the discrimination ability of the model. Hard example mining has been proved to be extraordinary beneficial for many tasks, e.g., person re-identification [19] and object detection [18]. However, previous works employ the hard example mining to select hard examples only from the current batch. As we know, the batch size is usually restricted due to the limited GPU memory. In this way, the scope of selecting hard pairs is only limited to the current batch, and the selected pairs are still not informative enough.

Contrastive Learning: Many contrastive learning methods are constructed based on the task of instance discrimination [66], [67], [68], [69], [70], where the different views of the same image are treated as positive pairs and different samples are treated as the negative pairs. In contrastive learning, each sample needs to be compared with many other samples to obtain promising performance. For example, Noise Contrastive Estimation [66] and Momentum Contrast [67], [70] employ a memory bank to collect more negative samples for comparison, and SimCLR [68] achieves this by using a large batch size. Our work follows a similar idea to extend the selecting space and select very hard sample pairs for training. However, many contrastive learning methods [67], [68], [69], [70] require a lot of computing resources (128 TPU v3 cores are taken in SimCLR [68]), while our work mainly study virtually increase the batch size with limited computing resources. Moreover, different from the contrastive loss that takes all samples to calculate the loss, our method only considers hard sample pairs and discards easy pairs.

III. The Proposed Method

In this section, we first give a simple overview of the proposed method. Then, we present the preliminaries of Triplet loss and Batch Hard Example Mining (BHEM). Later, we introduce two proposed algorithms in detail, namely Super-Batch and Cross-Batch Hard Example Mining(CBHEM).

A. Overview

Although the dataset of IvS face recognition usually contains a large amount of data, but each identity only contains two images. If we train the network on IvS dataset from scratch, the derived feature space would not be discriminative enough in the test stage. Thus, a two stage training method is employed for IvS face recognition. At first, the network is trained on MS-Celeb-1M [71] with AM-softmax [29], which helps the network to obtain a good initialization. Then, the network is finetuned on IvS dataset with a metric learning loss, where the classification layer is removed and the tremendous pressure on GPU resource is avoided. Considering the effectiveness of Triplet loss [15] for face recognition, it is adopted as the loss function in the metric learning stage. In the following, we will introduce Triplet loss and the proposed training strategies including S-Batch, VBS and CBHEM.

B. Triplet Loss

Triplet loss is computed based on a series of triplets, where each triplet consists of an anchor sample, a positive sample and a negative sample. Given a triplet $(\mathbf{x}_a, \mathbf{x}_{ap}, \mathbf{x}_{an})$ where $\mathbf{x}_{ap}, \mathbf{x}_{an}$ indicate the positive and negative samples with respect to the anchor sample $\mathbf{x}_a$, respectively. Formally, the triplet loss can be calculated as:

$$L_{tri}(\mathbf{x}_a, \mathbf{x}_{ap}, \mathbf{x}_{an}) = \left[\|\mathbf{x}_a - \mathbf{x}_{ap}\|_2 - \|\mathbf{x}_a - \mathbf{x}_{an}\|_2 + m\right]_+ \quad (1)$$

where $[z]_+ = \max(z, 0)$ and m represents the margin hyperparameter. $\|\mathbf{x}_1 - \mathbf{x}_2\|_2$ indicates the Euclidean distance between $\mathbf{x}_1$ and $\mathbf{x}_2$. Note that $\mathbf{x}_a$ and $\mathbf{x}_{ap}$ must come from a same identity but $\mathbf{x}_a$ and $\mathbf{x}_{an}$ come from different identities.

To make the training more effective, Batch Hrad Example Mining (BHEM) is used to only select hard triplets for training while dropping easy pairs. Assume each batch contains N samples, the Triplet loss with BHEM can be formulated as:

$$L_{batch} = \frac{1}{N}\sum_{a=1}^{N} L_{tri}\left(\mathbf{x}_a,\ \mathbf{x}_{ap}^{hard},\ \mathbf{x}_{an}^{hard}\right) \quad (2)$$

where $\mathbf{x}_{ap}^{hard}$ and $\mathbf{x}_{an}^{hard}$ indicate the hardest positive and negative samples from the current batch with maximum and minimum distances respect to the anchor sample $\mathbf{x}_a$, respectively. In this formulation, each sample would be set as the anchor sample, and it only selects the hardest positive and negative samples for training.

C. Training with Super Batch

The proposed Super Batch (S-Batch) aggregates many traditional batches together, and let it equivalent to train the network with a very large batch size. An illustration of the

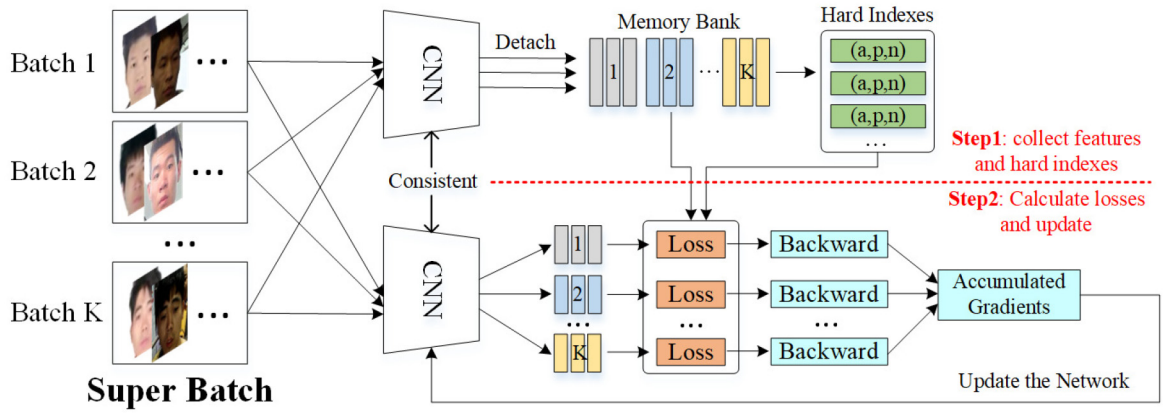


Fig. 3. An illustration of training IvS face recognition with the proposed Super-Batch. The proposed S-Batch aims to aggregate multiple traditional small batches to form a large batch. The proposed S-Batch is implemented in two steps. At first, the network extracts the features for all K traditional batches, and find the hard sample pairs across all batches. Second, the network computes the loss for each batch, and aggregates gradients from all batches together to update the network.

Algorithm 1: Training With S-Batch and VBS

Input: Training dataset $\mathcal{D} = \{\mathbf{I}_i, y_i\}$; Feature extraction network $\mathcal{F}(\cdot; \Theta)$; Learning rate η; The expansion times K in S-Batch; Various batch sizes ϕ.
Output: Feature extraction network $\mathcal{F}(\cdot; \Theta)$.

Initialize network $\mathcal{F}(\cdot; \Theta)$;
while *not convergence* **do**
  // step 1:
  $\nabla\Theta = 0$, $\tilde{\mathbf{X}}_{bank} = [\,]$, $\mathbf{Y}_{bank} = [\,]$;
  for $k = 1; k \leq K$ **do**
    Sampling images $\mathbf{I} = [\mathbf{I}_1, \cdots, \mathbf{I}_N]$ and labels $\mathbf{Y} = [y_1, \cdots, y_N]$;
    Extract batch features $\mathbf{X} = \mathcal{F}(\mathbf{I}; \Theta)$;
    $\tilde{\mathbf{X}} = detach(\mathbf{X})$, $\tilde{\mathbf{X}}_{bank} = [\tilde{\mathbf{X}}_{bank}, \tilde{\mathbf{X}}]$;
  end
  Selecting hard triplets H_{index} based on $\tilde{\mathbf{X}}_{bank}$ and $\mathbf{Y}_{bank}$ according to various batch sizes ϕ;
  for $k = 1; k \leq K$ **do**
    Sampling images $\mathbf{I} = [\mathbf{I}_1, \cdots, \mathbf{I}_N]$;
    Extract batch features $\mathbf{X} = \mathcal{F}(\mathbf{I}; \Theta)$;
    Calculate L^{ϕ}_{VBS} according to Eq. (6) and H_{index};
    $\nabla\Theta \leftarrow \nabla\Theta + \cdot\frac{\partial L^{\phi}_{VBS}}{\partial\Theta}$;
  end
  Update $\Theta \leftarrow \Theta - \eta\nabla\Theta$ with SGD;
end

proposed S-Batch is shown in Fig. 3 and the corresponding algorithm is shown in Algorithm 1. Assume the dataset is denoted as $\mathcal{D} = \{\mathbf{I}_i, y_i\}$, where $\mathbf{I}_i$ and y_i indicate the i^{th} image and its corresponding label. The proposed S-Batch strategy can be divided into two steps. At first, we sample K batches of data where K refers to the expansion times of S-Batch to the traditional batch. We first let all samples forward pass the network $\mathcal{F}(\cdot; \Theta)$ with Θ denoting the network parameters, and collect all features and their labels to the memory bank $\tilde{\mathbf{X}}_{bank}$ and $\mathbf{Y}_{bank}$, respectively. Note all features in $\tilde{\mathbf{X}}_{bank}$ are detached from the computational graph with only numerical values, which let them occupy a very little GPU memory.

In the second step, we mainly aim to calculate the loss function and update the network. We first analyze the loss function of training with a real large batch size $K \times N$, which can be formulated as:

$$L_{batch} = \frac{1}{KN}\sum_{a=1}^{KN} L_{tri}\left(\mathbf{x}_a, \mathbf{x}_{ap}^{hard}, \mathbf{x}_{an}^{hard}\right) \tag{3}$$

where very hard triplets $\{\mathbf{x}_a, \mathbf{x}_{ap}^{hard}, \mathbf{x}_{an}^{hard}\}$ can be captured from the large batch set of $K \times N$ images. This loss function is impossible to be implemented in practical training with limited GPU memory. Let's look at this problem from a different way. As we have mentioned above, the features and labels for the whole large batch have been collected. Thus, all the hardest positive and negative samples for each anchor from a large set can be founded out based on the collected information. In other words, the hard triplets $\{\mathbf{x}_a, \mathbf{x}_{ap}^{hard}, \mathbf{x}_{an}^{hard}\}$ can be selected out based on $\tilde{\mathbf{X}}_{bank}$ and $\mathbf{Y}_{bank}$. As we know, Triplet loss is calculated based on triplets, but for each triplet, all three samples will not appear in a same traditional batch. Thus, we can divide the triplet loss into three parts, with each part corresponding to a sample and calculated in a specific batch. Thus, we can rewrite Eq. (3) as:

$$L_{batch} = \frac{1}{KN}\sum_{a=1}^{KN}\Big(L_{tri}\left(\mathbf{x}_a, \tilde{\mathbf{x}}_{ap}^{hard}, \tilde{\mathbf{x}}_{an}^{hard}\right) + L_{tri}\left(\tilde{\mathbf{x}}_a, \mathbf{x}_{ap}^{hard}, \tilde{\mathbf{x}}_{an}^{hard}\right) + L_{tri}\left(\tilde{\mathbf{x}}_a, \tilde{\mathbf{x}}_{ap}^{hard}, \mathbf{x}_{an}^{hard}\right)\Big) \tag{4}$$

where $\tilde{\mathbf{x}}_a, \tilde{\mathbf{x}}_{ap}^{hard}, \tilde{\mathbf{x}}_{an}^{hard}$ are the features in $\tilde{\mathbf{X}}_{bank}$, and no gradient will be backpropagated along with those variables. Eq. (3) and Eq. (4) are equivalent in network training because their gradients on $\mathbf{x}_a$, $\mathbf{x}_{ap}^{hard}$ and $\mathbf{x}_{an}^{hard}$ are consistent. To let Eq. (4) can be successfully applied to our S-Batch, we further rewrite it as:

$$\begin{aligned} L_{s-batch}^{K} &= \frac{1}{K}\sum_{k=1}^{K} L_{s-batch}^{K,B_k} \\ L_{s-batch}^{K,B_k} &= \frac{1}{N}\sum_{a=1}^{N} 1_{\mathbf{x}_a}^{B_k} L_{tri}\left(\mathbf{x}_a, \tilde{\mathbf{x}}_{ap}^{hard}, \tilde{\mathbf{x}}_{an}^{hard}\right) \\ &\quad + 1_{\mathbf{x}_{ap}^{hard}}^{B_k} L_{tri}\left(\tilde{\mathbf{x}}_a, \mathbf{x}_{ap}^{hard}, \tilde{\mathbf{x}}_{an}^{hard}\right) \\ &\quad + 1_{\mathbf{x}_{an}^{hard}}^{B_k} L_{tri}\left(\tilde{\mathbf{x}}_a, \tilde{\mathbf{x}}_{ap}^{hard}, \mathbf{x}_{an}^{hard}\right) \end{aligned} \tag{5}$$

where $1_{\mathbf{x}_a}^{B_k}$ is an indicator function. $1_{\mathbf{x}_a}^{B_k} = 1$ if $\mathbf{x}_a$ is contained in the k^{th} batch B_k; otherwise $1_{\mathbf{x}_a}^{B_k} = 0$. In this way, the whole loss as shown in Eq. (5) can divided into K sub-losses with each sub-loss $L_{s-batch}^{K,B_k}$ calculated for the data of each batch. To the end, we train the network with virtually setting a very large batch size by using S-Batch strategy, which helps the network to capture more harder triplets from a very large image set.

Various Batch Sizes: Consider an extreme example, the batch size is set to be equal with the dataset size. For each anchor sample, it hardest negative sample is often the same image, where the diversity of select hard triplets is reduced. To trade off the difficulty and diversity of the selected triplets, we propose a strategy named Various Batch Sizes (VBS) in S-Batch, where the hard triplets of different batch sizes can be selected for training concurrently. The corresponding loss function can be written as:

$$L_{VBS}^{\phi} = \sum_{\varphi \in \phi} L_{s-batch}^{\varphi} \tag{6}$$

where ϕ is a set of values indicating the different batch sizes. In our experiments, the maximal φ in ϕ is usually set to equal to the expansion times K in our S-Batch. For example, if we set $K = 10$, we can use a VBS of $\phi = \{1, 5, 10\}$ for training. For each anchor sample, three triplets will be selected from the S-Batches with the batch size of $1 \times N$, $5 \times N$ and $10 \times N$, respectively. For convenience, every φ traditional batches in order would be regarded as a group to select the hard triplets of batch scale φ. For example, for $\phi = \{1, 5, 10\}$, batches $1, \ldots, 5$ would be regarded as a group and batches $6, \ldots, 10$ would be regarded as the other group to select hard triplets at the scale $\varphi = 5$.

Indexing From a Table: We find out all hardest positive and negative samples for each anchor sample across the whole S-Batch based on $\tilde{\mathbf{X}}_{bank}$ and $\mathbf{Y}_{bank}$. To improve the efficiency, all indexes of those selected hard samples would be stored to a table H_{index}. In the loss calculating stage, the hard triplets for each batch can be obtained by searching the index table H_{index}. In this way, repeated searches are avoided and thus the efficiency can be improved.

D. Cross-Batch Hard Example Mining

According to previous works [20], [21], the historical features also can be used as references for selecting hard sample pairs for training. To achieve this, we use three queues, namely $Q_{\mathbf{X}}$, $Q_{\mathbf{I}}$ and $Q_{\mathbf{Y}}$, to collect the features, images and labels of the historical iterations, respectively. More specifically, given a batch (also could be a S-Batch) of images $\mathbf{I}$ and their labels $\mathbf{Y}$, we first employ the network $\mathcal{F}(\cdot; \Theta)$ to extract the corresponding features $\mathbf{X}$. Then, the loss will be calculated for this batch to update the network. Please note that the loss is calculated according to Section III-B if taking the traditional batch, or according to Section III-C if taking the S-Batch. Then, the raw images $\mathbf{I}$, the labels $\mathbf{Y}$ and the features $\mathbf{X}$ are collected by the queues $Q_{\mathbf{X}}$, $Q_{\mathbf{I}}$ and $Q_{\mathbf{Y}}$, respectively. We set the length of queues as MN (MKN if S-Batch is taken), which indicates that the nearby M batches of samples would be collected by those queues, with the current iteration's enqueued and the oldest iteration's dequeued. Then, a further hard example mining can be conducted based on $Q_{\mathbf{X}}$ and $Q_{\mathbf{I}}$. In the mining stage, we first take a comparison on all positive sample pairs, and only a small proportion of most difficult positive pairs (with the maximum distances) will be selected out. The proportion is denoted as r, and we set it as 0.2 in our experiments. Then, for each pair of the selected positive pairs, we randomly select a sample as the anchor sample, which is used to compare with all samples in $Q_{\mathbf{X}}$, and select the hardest one as the negative sample (with a minimum distance). Those hard triplets across different iterations are denoted as cross-batch hard triplets, which are further collected by using a queue $Q_{\mathbf{I}}^{hard}$. When the length of the queue $Q_{\mathbf{I}}^{hard}$ reaches or exceeds the batch size, the corresponding images in queue $Q_{\mathbf{I}}^{hard}$ of the selected hard triplets would be taken out and fed to the network for training again. The loss for those selected hard triplets is calculated according to Eq. (1). An illustration of the above training strategy is shown in Fig. 4.

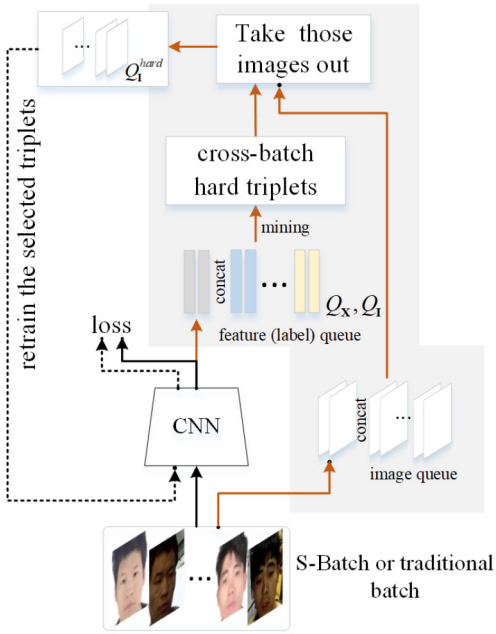


Fig. 4. An illustration of our CBHEM. In CBHEM, we collect features, labels and images of past batches by queues $\mathcal{Q}_{\mathbf{X}}$, $\mathcal{Q}_{\mathbf{Y}}$ and $\mathcal{Q}_{\mathbf{I}}$. We then select cross-batch hard triplets based on those queues. Finally, we take the raw images of select hard triplets out, and input them to the network for training again.

Note that there is no conflict between data collection and network updating. For each batch of data, the loss calculation is also conducted based on the extracted features to update the network. Moreover, the proposed CBHEM also can be used concurrently with S-Batch, which further expands the selecting space to select more harder triplets for training. The network's discriminative capability also can be further enhanced.

IV. Experiments

In this section, we first introduce datasets we have employed in our experiments. Then, we conduct a parameter analysis to select the best values for hyper-parameters, and conduct ablation studies to analyze the contributions for each component.

Finally, we compare the proposed method to prior arts and make some deep analysis and discussions.

A. Datasets

All networks are first pretrained on MS-Celeb-1M, and then finetuned on Private IvS Dataset or MegaFace-Bisample [2]. Then, the model is evaluated on Private-IvS Dataset, Public-IvS dataset [2] and LFW-BLUFR [72], [73]. Since MS-Celeb-1M is a well-known dataset in face recognition, we omit its introduction here and we mainly introduce other datasets in this section.

Private-IvS: The dataset is collected from practical face verification systems. Each identity in this dataset has two images, namely one ID photo and one spot photo. This dataset is divided into four subsets, namely Private-IvS-Train-L(large), Private-IvS-Train-S(mall), Private-IvS-Val and Private-IvS-Test, with containing 2 million, 500,000, 5,000 and 5,000 identities, respectively. Note Private-IvS-Train-S is a subset of Private-IvS-Train-L. All ID-Spot pairs in Private-IvS-Val and Private-IvS-Test are employed for validation and evaluation, respectively.

Public-IvS: This is a public dataset for IvS face recognition evaluation. The data are crawled from the internet, and then manually cleaned. The dataset contains 5,507 images of 1,262 identities in total. The model is evaluated by all ID-spot pairs of any two images during testing stage. Note that this dataset is collected from the internet, and thus it contains some noises of wrong annotations although this dataset is manually cleaned.

Megaface-bisample and LFW-BLUFR: Following Megaface-bisample protocol in the work [2], we also train the model with the open MF2 dataset [74] and evaluate it on LFW [72] with BLUFR protocol [73]. MF2 dataset contains 657,559 identities in total, but only two samples are randomly selected for each identity to mimic the bisample data for training. More details about this protocol can be founded in the work [2].

B. Settings and Metrics

All faces are detected and then aligned by five landmarks (including two eyes, nose tip and two mouth corners). The network takes the input of RGB image with the size of 120×120. Random flipping is employed for data augmentation in the training stage. The network is trained based on Pytorch by Stochastic Gradient Descent (SGD) optimizer with a weight decay of 0.0005 and a momentum of 0.9. Following the work [2], a 64-layer residual network is adopted as our backbone. The learning rate starts from 0.01 and is reduced by a factor of 10 along with the number of iterations increases. The network is trained on three NVIDIA GTX 2080Ti GPUs in parallel with a batch size of 240. The features extracted based on the raw image and its flipped copy are concatenated together as final face features. The ROC curve is employed as the evaluating metric according to the work [2], and the pair score is calculated based on cosine distance.

TABLE I
THE ANALYSIS OF K IN S-BATCH ON PRIVATE-IVS-VAL DATASET. THE BEST RESULTS ARE HIGHLIGHTED

K	True Positive Rate (TPR)		
	FPR=1e-6	FPR=1e-5	FPR=1e-4
N/A	75.06	87.52	94.72
5	80.88	90.18	96.28
10	**81.89**	**91.24**	**96.90**
20	81.24	90.54	96.72
40	78.06	89.40	95.30

TABLE II
THE ANALYSIS OF ϕ IN S-BATCH ON PRIVATE-IVS-VAL DATASET. THE BEST RESULTS ARE HIGHLIGHTED

ϕ	True Positive Rate (TPR)		
	FPR=1e-6	FPR=1e-5	FPR=1e-4
N/A	75.06	87.52	94.72
10	81.89	91.24	96.90
5, 10	88.44	94.28	97.80
1, 5, 10	**88.96**	**94.80**	**97.90**

C. Parameter Analysis

In this section, extensive experiments are conducted with different parameter settings. 'N/A' denotes the network is only trained by BHEM of Triplet loss. It is also taken as the baseline for comparisons to show how much performance each setting can be improved.

Expansion Times K in S-Batch: Experimental results with various K are shown in Table I. Training with a large K can achieve better performance, where harder triplets can be selected for training. For example, the network achieves the highest performance with $K = 10$. When further increasing K from 10 to 40, the performance is reduced because of the degeneration of variety in selected hard triplets where larger K is employed. From the experimental results, we can see that although larger K can help to capture harder sample pairs, too large K does not necessarily lead to better performance. Using a large K sacrifices the diversity of samples and also reduces the training efficiency. Therefore, choosing an appropriate K would contribute to ensuring its performance to optimize most. Thus K is set to 10 in our following experiments.

Various Batch Sizes ϕ in S-Batch: Based on the above analysis (setting $K = 10$), we further conduct the experiments with various ϕ settings, including $\{10\}$, $\{5, 10\}$ and $\{1, 5, 10\}$. The experimental results are shown in Table II. Employing various batch sizes really improves the network's discriminative capability by enhancing the variety of selected hard triplets. The network reaches the highest performance with $\phi = \{1, 5, 10\}$, which shows that the triplets would be selected from three kinds of batch sizes in S-Batch, i.e., $1 \times N$, $5 \times N$ and $10 \times N$. Intuitively, compared with using a single setting of S-Batch, the selected triplets would be more abundant from various difficulties. and this setting would be used in our following experiments.

Crossed Batches M in CBHEM: Larger M can obtain a larger selecting space for each anchor sample, and helps the network to choose harder samples. However, if a very large M is employed, the old features in the memory bank are slightly effective and lots of computations will be brought when selecting the cross-batch hard triples. To select the best

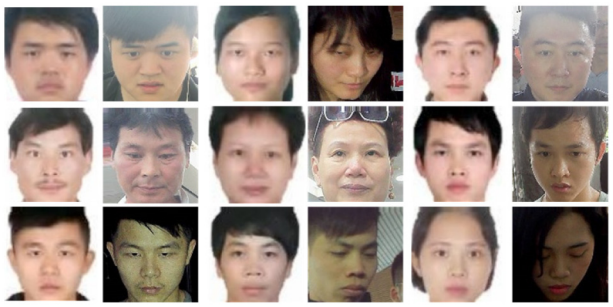


Fig. 5. Examples of Private-IvS dataset. Each identity contains a spot image and a ID image. The ID face is captured in a constrained environment (with frontal face and clean background), while the spot face is captured in an unconstrained environment (with large variations in head poses, illuminations, backgrounds and so on).

value for M, we conduct the experiments with various settings. As shown in Table III, only a very little performance improvement can be achieved when setting a M greater than 10. using $M = 10$ can reach the highest accuracy at False Positive Rate (FPR) = 1e-6. Although using $M = 20$ or 40 can achieve a little bit performance improvement at FPR=1e-5 and FPR=1e-4, lot of computations will be brought compared with using $M = 10$ where it needs to compare with more features to select hard sample pairs for each anchor. Thus, we use $M = 10$ in the following experiments in consideration of model's efficiency.

D. Ablation Studies

In this section, ablation studies are conducted to analyze the proposed components including S-Batch, VBS and CBHEM. The network trained with Triplet loss and BHEM is employed as the baseline method, and we denote it as 'Tri'. The experiments are evaluated on two sets, namely Private-IvS-Val and Private-IvS-Test, and the experimental results are shown in Fig. 6 (a) and (b), respectively. All components can steadily improve the performance on both Private-IvS-Val and Private-IvS-Test datasets, and the best performance can be achieved when all components are employed (denoted as 'Tri + S-Batch(VBS)+CBHEM'). More specifically, both CBHEM and S-Batch improve the true positive rate at FPR = 1e-6 by about 2%, but the VBS strategy can further improve that by about 5%. This shows that increasing the diversity of selected triplets is very important and necessary to improve the performance, especially employing a large batch size for training. All those components are proposed to select effective triplets for training, by extending the selecting space or increasing their diversity. The improvements demonstrate the effectiveness of the proposed components.

E. Comparisons to Prior Arts

Results on Private-IvS-Test: The baseline method of Triplet loss with BHEM is taken as the baseline (we denote it as 'Tri'). Moreover, we also train the network with some popular metric learning losses, like Angular Loss [53], Lifted structure [50], MS loss [55] and N-pair [52]. Besides, we also take AM-softmax [29], ArcFace [16] and ElasticFace [75]

TABLE III
THE ANALYSIS OF M IN CBHEM ON PRIVATE-IVS-VAL DATASET. THE TOP-2 RESULTS ARE HIGHLIGHTED

M	True Positive Rate (TPR)		
	FPR=1e-6	FPR=1e-5	FPR=1e-4
N/A	75.06	87.52	94.72
5	77.32	89.22	95.56
10	**78.12**	**89.70**	**95.78**
20	78.50	89.72	95.74
40	**78.82**	**89.98**	**95.92**

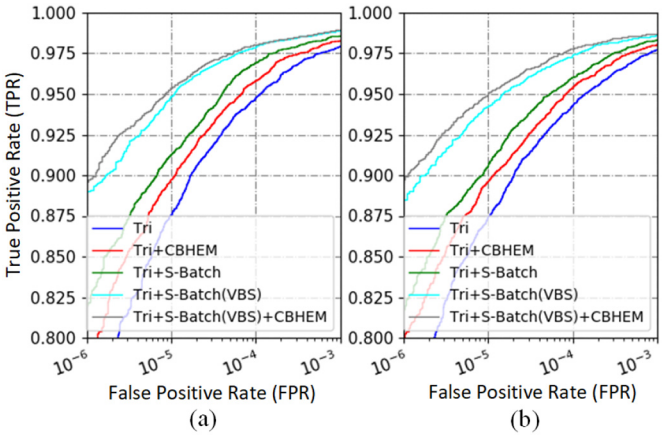


Fig. 6. The ablation studies on Private-IvS-Val (left) and Private-IvS-Test (right) with training on Private-IvS-Train-S.

TABLE IV
THE COMPARISONS ON PRIVATE-IVS-TEST DATASET. THE TOP-2 RESULTS ARE HIGHLIGHTED

Method	True Positive Rate (TPR)		
	FPR=1e-6	FPR=1e-5	FPR=1e-4
AM-softmax$^{\diamond}$ [29]	45.46	62.10	78.28
ElasticFace$^{\circ}$ [75]	50.28	68.48	83.42
ArcFace$^{\diamond}$ [16]	68.46	82.02	91.74
Angular* [53]	82.36	90.92	95.82
Lifted* [50]	82.76	91.92	96.38
MS Loss* [55]	78.80	89.16	95.10
N-pair* [52]	77.38	87.84	94.84
Tri*	76.08	87.36	94.26
Tri†	80.02	89.70	95.42
Ours*	**89.54**	**95.02**	**97.80**
Ours†	**91.40**	**96.20**	**98.12**

training on * Private-IvS-Train-S, † Private-IvS-Train-L, $^{\diamond}$ MS-Celeb-1M and $^{\circ}$ Glint360K.

for comparison. Those three methods are classification-based methods, which are hardly trained on the IvS dataset with massive identities. Therefore, we report their performance of training on MS-Celeb-1M or Glint360K datasets. Due to limited computing resources, those losses are trained on Private-IvS-Train-S. As shown in Table IV and Fig. 7 (a), our method outperforms all previous methods. For the baseline method 'Tri', our method outperforms it by a large margin, which clearly demonstrates the effectiveness of the proposed components. For 'Tri' and our proposed method, the network trained on Private-IvS-Train-L can achieve better performance than that on Private-IvS-Train-S. This shows using more training samples can boost the performance. Whether training on Private-IvS-Train-S or Private-IvS-Train-L, the proposed method outperforms the baseline method

TABLE V
THE COMPARISONS ON PUBLIC-IVS DATASET. THE TOP-2 RESULTS ARE HIGHLIGHTED

Method	True Positive Rate (TPR)		
	FPR=1e-5	FPR=1e-4	FPR=1e-3
COTS-1 [1]	83.78	89.92	92.90
COTS-2 [1]	94.74	97.03	97.88
CenterFace [7]	35.97	53.30	69.18
SphereFace [28]	53.21	69.25	83.11
DocFace+ [1]	91.88	96.48	98.40
LBL(DPS) [2]	93.62	97.21	98.83
Angular* [53]	93.00	96.82	98.71
Lifted* [50]	89.48	95.28	98.42
MS Loss* [55]	90.43	95.77	98.38
N-pair* [52]	87.82	94.09	97.66
Tri*	76.08	87.36	94.26
Tri†	80.02	89.70	95.42
Ours*	**95.62**	**98.11**	**98.95**
Ours†	**96.65**	**98.23**	**99.06**

training on * Private-IvS-Train-S and † Private-IvS-Train-L.

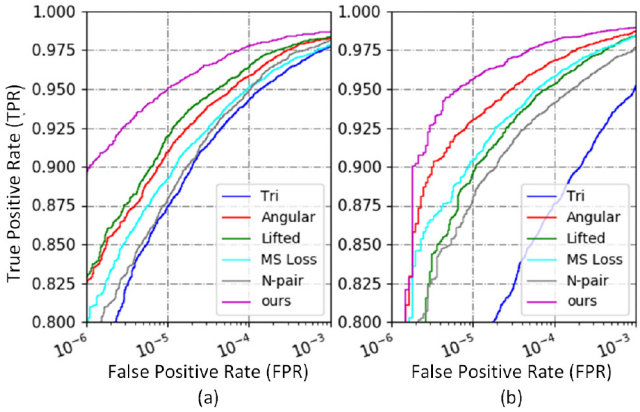


Fig. 7. Comparisons on (a) Private-IvS-Test and (b) Public-IvS datasets. All models are trained with Private-IvS-Train-S.

by a large margin, which clearly demonstrates the effectiveness of the proposed components. For example, our approach improves performance by 13.46% and 11.38% when training with Private-IvS-Train-S or Private-IvS-Train-L, respectively. Note that Private-IvS-Train-L contains two million identities, and those promising improvements show our method can deal with the dataset with a very large number of classes very well.

Results on Public-IvS: The compared methods including COTS-1 & COTS-2 [1], CenterFace [7], SphereFace [28], DocFace+ [1] and LBL(DPS) [2]. Besides, we also implement Angular Loss [53], Lifted structure [50], MS loss [55] and N-pair [52] for comparisons. As shown in Table V and Fig. 7 (b), Compared with the baseline method, i.e., 'Tri', the performance has been dramatically improved from 76.08% to 95.62% on Private-IvS-Train-L, respectively. Our method outperforms all compared works and achieves the state-of-the-art performance with reaching a highest VR of 96.65% at FPR = 1e-5. Compared with previous best method LBL(DPS), our method improve the accuracy by 3.03%, 1.02% and 0.23% at FPR = 1e-5, FPR = 1e-4 and FPR = 1e-3, respectively. Those improvements are very considerable although the percentages of performance improvement looks not large because the compared method, i.e., LBL(DPS), has reached

TABLE VI
THE COMPARISONS ON LFW-BLUFR FOLLOWING THE MEGAFACE-BISAMPLE. THE BEST RESULTS ARE HIGHLIGHTED

Method	True Positive Rate (TPR)		
	FPR=1e-5	FPR=1e-4	FPR=1e-3
Angular [53]	73.05	95.27	98.01
Lifted Struct [50]	53.45	75.46	90.50
MS Loss [55]	72.27	89.90	94.54
N-pairs [52]	50.30	73.40	90.16
LBL(DPS) [2]	**73.86**	88.03	95.68
Ours	72.52	**96.02**	**98.13**

Fig. 8. Examples of False Accept and False Reject on Private-IvS-Test datasets.

the accuracies of 93.62%, 97.21% and 98.83% at FPR = 1e-5, FPR = 1e-4 and FPR = 1e-3, respectively. Moreover, our method still can outperform all previous works with only using 500 thousand identities for training (i.e., Private-IvS-Train-S).

Results on LFW-BLUFR: We compare our method with Lifted Struct [50], N-pairs [52] and LBL(DPS) [2]. The results reported in the work [2] of Lifted Struct and N-pairs are taken for comparisons. The comparisons are shown in Table VI. Our proposed method achieves the best performance on FPR = 1e-4 and FPR = 1e-3, with outperforming previous best results by 7.99% and 2.45%, respectively. The proposed method didn't achieve the best result at FPR = 1e-5 may be due to the TPR at low FPR is usually unstable and it is easy to be affected by noises.

F. Discussions

Visual Assessment: Generally, our model can perform well for IvS face recognition except for some extreme cases. We show some examples of False Accept and False Reject of Private-IvS-Test and Public-IvS datasets in Fig. 8 and Fig. 9, respectively. For Private-IvS-Test, failures mainly come from extreme pose, poor illuminations, big age gap and so on. For Public-IvS, it contains some noises because its images are collected from the internet, and some failures come from wrong annotations. Thus, besides the external factors mentioned above, labeling noises also result to the 'failures'.

Experiments With Classical Contrastive Loss: Triplet loss and Contrastive loss are two loss functions that are often used with hard example mining. Therefore, we employ Contrastive

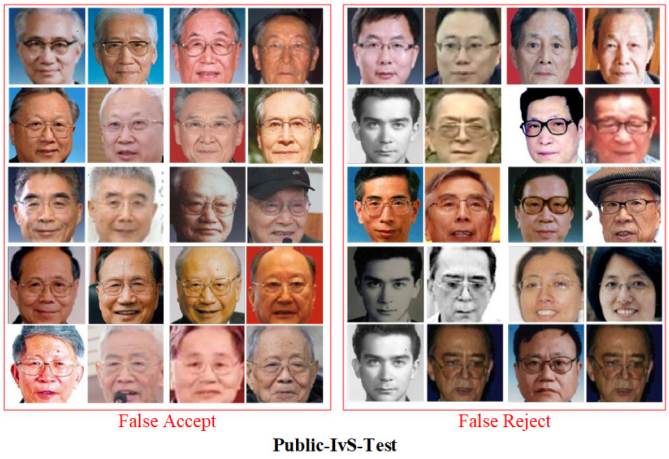


Fig. 9. Examples of False Accept and False Reject on Public-IvS datasets. Note that this dataset is collected from the internet and contains noises of wrong annotations. For example, for the first two rows of 'False Accept' part, the two images in each pair come from a same identity but the dataset gives different identity labels to them.

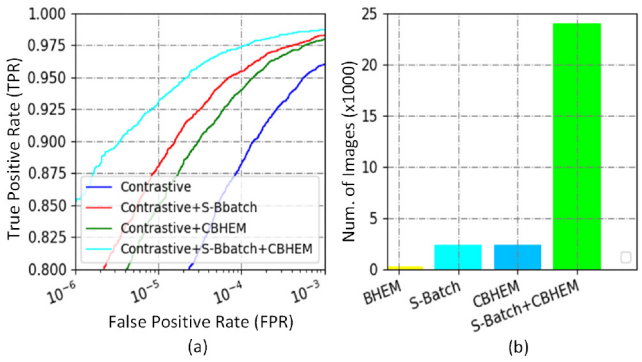


Fig. 10. (a) Experimental results with contrastive loss. (b) Comparisons of the selecting space size.

loss to take some additional experiments, which further validate the effectiveness of our proposed S-Batch and CBHEM. As shown in Fig. 10 (a), both S-Batch and CBHEM can significantly improve the performance when assembling them with the Comparative loss. For example, the true positive rate at FPR=1e-4 has been improved by more than 5% when using S-Batch or CBHEM. The model achieves the highest performance when both components are employed together, where the true positive rate at FPR=1e-6 reaches to 85%. The improvements further show the effectiveness of our methods, and also show their good generalization ability.

Discussions With InfoNCE Loss: In contrastive learning [67], [70], InfoNCE [76] is usually adopted as the loss function. In this section, we also compare the proposed method with InfoNCE loss. According to previous works [67], [70], a memory bank is also employed to collect additional samples to achieve better performance We set the memory bank to contain 2 batches of features, which is also the best setting of achieving highest performance. The experimental results are shown in Table VII, where the proposed method outperforms InfoNCE loss by a considerable margin. The reason may come from the following aspects. At first, InfoNCE treats all sample pairs equally while the proposed method only selects the hardest sample pairs for training. However, face recognition

TABLE VII
THE COMPARISONS ON PRIVATE-IVS-TEST DATASET. ALL MODELS ARE PRETRAINED ON MS-CELEB-1M AND FINETUNED ON PRIVATE-IVS-S DATASET. THE TOP PERFORMANCE IS HIGHLIGHTED

Method	True Positive Rate (TPR)		
	FPR=1e-6	FPR=1e-5	FPR=1e-4
Triplet Loss	76.08	87.36	94.26
InfoNCE	74.58	87.76	95.08
Ours	89.54	95.02	97.80

is largely different from the general object recognition, where face datasets contain a large number of identities but have very small inter-class variations. and thus treating all pairs equally may hardly enlarge inter-class distances, Taking the easy sample pairs for training would damage the performance of the model. Second, although the contrastive learning [67], [70] employs a memory bank to collect more samples for calculating losses, the learning is biased where the gradients will not be returned to the samples in the memory bank. The biased learning increases the difficulty of network convergence, and the network is easy to collapse when using a large memory bank (e.g., the training loss becomes 'nan' during training when the memory bank contains 8 batches of samples). All in all, the experimental results further verify the effectiveness of the proposed method.

Analysis of Selecting Space: As shown in Fig. 10 (b), when setting the batch size as 240, the proposed method can extend the selecting pace to a large set with 24,000 images, which is 100 times larger than the traditional batch. This shows that our method helps the network to select harder sample pairs for training compared with only using BHEM, and also helps the network to capture more effective features. Our method greatly improves the effectiveness of hard examples mining by extending the selecting space under the limited GPU resources.

Discussions on Future Works: In this paper, we have proposed a Super-Batch and CBHEM to virtually increase the batch size and select very hard sample pairs for improving the performance. However, such selection does not always work when the dataset contains noise, where the noisy sample would be easily treated as the hardest positive or negative samples (see failure cases in Section IV-F). Fortunately, the noise mainly exists in Public-IvS dataset, which is a small dataset for evaluation. For Private-IvS dataset, it is collected from real-world applications and only contains less noise. Even though, how to accurately select hard sample pairs against noisy labels is still an important problem needed to be studied in the future especially when applying the proposed method to some coarsely collected datasets.

V. CONCLUSION

In this paper, we aim to improve network's discriminative capability of IvS face recognition by choosing very hard triplets and increasing the diversity of selected hard triplets. We extend the selecting space mainly from the following two aspects. 1) we propose a S-batch which combines multiple traditional batches to a large batch. 2) historical features are also used as a reference to find cross-batch hard triplets. Besides, we also propose a VBS to select the hard triplets from different batch scales, which increases the diversity of selected

hard triplets. With those strategies, our approach achieves the state-of-the-art performance on multiple benchmark datasets including Private-IvS, Public-IvS and LFW-BLUFR.

REFERENCES

[1] Y. Shi and A. K. Jain, "DocFace+: ID document to selfie matching," *IEEE Trans. Biometrics Behav. Identity Sci.*, vol. 1, no. 1, pp. 56–67, Jan. 2019.

[2] X. Zhu *et al.*, "Large-scale bisample learning on ID versus spot face recognition," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 684–700, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[4] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[5] N. Zhu, "Neural architecture search for deep face recognition," 2019, *arXiv:1904.09523*.

[6] Y. Zhang *et al.*, "Global-local GCN: Large-scale label noise cleansing for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7728–7737.

[7] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*, 2016, pp. 499–515.

[8] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "Adaptiveface: Adaptive margin and sampling for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11947–11956.

[9] Y. Huang *et al.*, "CurricularFace: adaptive curriculum learning loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5901–5910.

[10] X. Wang, S. Wang, C. Chi, S. Zhang, and T. Mei, "Loss function search for face recognition," in *Proc. ICML*, 2020, pp. 10029–10038.

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

[12] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VggFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, 2018, pp. 67–74.

[13] X. An *et al.*, "Partial FC: Training 10 million identities on a single machine," 2020, *arxiv.2010.05222*.

[14] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Self-restrained triplet loss for accurate masked face recognition," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108473.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019, pp. 4690–4699.

[17] Y. Shi and A. K. Jain, "DocFace: Matching ID document photos to selfies," in *Proc. IEEE BTAS*, 2018, pp. 1–8.

[18] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.

[19] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[20] X. Wang, H. Zhang, W. Huang, and M. R. Scott, "Cross-batch memory for embedding learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6388–6397.

[21] T. Zichang *et al.*, "Cross-batch hard example mining with pseudo large batch for ID vs. spot face recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 3224–3235, 2022.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[24] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[25] H. Zhang *et al.*, "ResNest: Split-attention networks," 2020, *arXiv:2004.08955*.

[26] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[27] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," 2014, *arXiv:1406.4773*.

[28] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, 2017, pp. 6738–6746.

[29] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. CVPR*, 2018, pp. 5265–5274.

[30] Z. Zhu *et al.*, "WebFace260m: A benchmark unveiling the power of million-scale deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10492–10502.

[31] P. Li, B. Wang, and L. Zhang, "Virtual fully-connected layer: Training a large-scale face recognition dataset with limited computational resources," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13315–13324.

[32] W. Liu *et al.*, "Learning towards minimum hyperspherical energy," in *Proc. NeurIPS*, 2018, pp. 6225–6236.

[33] Y. Wu *et al.*, "Rotation consistent margin loss for efficient low-bit face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6865–6875.

[34] Y. Duan, J. Lu, and J. Zhou, "UniformFace: Learning deep equidistributed representation for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3415–3424.

[35] W. Liu, Y. Wen, B. Raj, R. Singh, and A. Weller, "Sphereface revived: Unifying hyperspherical face recognition," 2021, *arXiv:2109.05565*.

[36] Y. Wen, W. Liu, A. Weller, B. Raj, and R. Singh, "SphereFace2: Binary classification is all you need for deep face recognition," 2021, *arXiv:2108.01513*.

[37] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, "Attentional feature-pair relation networks for accurate face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5472–5481.

[38] Q. Wang, T. Wu, H. Zheng, and G. Guo, "Hierarchical pyramid diverse attention networks for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8326–8335.

[39] Q. Wang and G. Guo, "AAN-Face: Attention augmented networks for face recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 7636–7648, 2021.

[40] Q. Wang and G. Guo, "LS-CNN: Characterizing local patches at multiple scales for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1640–1653, 2019.

[41] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[42] X. Hou, Y. Li, and S. Wang, "Disentangled representation for age-invariant face recognition: A mutual information minimization perspective," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3692–3701.

[43] Z. Huang, J. Zhang, and H. Shan, "When age-invariant face recognition meets face age synthesis: A multi-task learning framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7282–7291.

[44] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1415–1424.

[45] J. Zhao *et al.*, "Towards pose invariant face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2207–2216.

[46] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, "Co-mining: Deep face recognition with noisy labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9358–9367.

[47] Y. Zhang *et al.*, "Adaptive label noise cleaning with meta-supervision for deep face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15065–15075.

[48] H. Qiu, B. Yu, D. Gong, Z. Li, W. Liu, and D. Tao, "SynFace: Face recognition with synthetic data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10880–10890.

[49] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, 2006, pp. 1735–1742.

[50] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012.

[51] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," 2016, *arXiv:1611.00822*.

[52] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. NeurIPS*, 2016, pp. 1849–1857.

[53] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2593–2601.

[54] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 360–368.

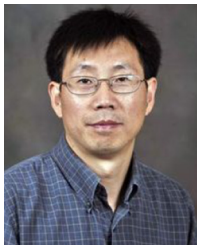
[55] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5022–5030.
[56] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5207–5216.
[57] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3238–3247.
[58] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 403–412.
[59] Z. Yang, M. Bastan, X. Zhu, D. Gray, and D. Samaras, "Hierarchical proxy-based loss for deep metric learning," in *Proc. WACV*, 2022, pp. 449–458.
[60] H. Xuan, R. Souvenir, and R. Pless, "Deep randomized ensembles for metric learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 723–734.
[61] P. Jacob, D. Picard, A. Histace, and E. Klein, "Metric learning with horde: High-order regularizer for deep embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6539–6548.
[62] W. Ge, "Deep metric learning with hierarchical triplet loss," in *Proc. ECCV*, 2018, pp. 272–288.
[63] K. Roth, T. Milbich, and B. Ommer, "PADS: Policy-adapted sampling for visual similarity learning," in *Proc. CVPR*, 2020, pp. 6567–6576.
[64] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2780–2789.
[65] Y. Zhao, Z. Jin, G.-J. Qi, H. Lu, and X.-S. Hua, "An adversarial approach to hard triplet generation," in *Proc. ECCV*, 2018, pp. 508–524.
[66] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.
[67] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[68] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[69] J.-B. Grill *et al.*, "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21271–21284.
[70] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.
[71] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-CELEB-1M: A dataset and benchmark for large-scale face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 87–102.
[72] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Rep. 07-49, Oct. 2007.
[73] S. Liao, Z. Lei, D. Yi, and S. Z. Li, "A benchmark study of large-scale unconstrained face recognition," in *Proc. IEEE Int. Joint Conf. Biometrics*, 2014, pp. 1–8.
[74] A. Nech and I. Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7044–7053.
[75] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "ElasticFace: Elastic margin loss for deep face recognition," 2021, *arXiv:2109.09416*.
[76] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arxiv.abs/1807.03748*.

Zichang Tan (Member, IEEE) received the B.E. degree from the Department of Automation, Huazhong University of Science and Technology in 2016, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 2021. Since July 2021, he has been with Baidu as a Researcher. His main research interests include deep learning, computer vision, and biometrics in particular. He was named as an outstanding graduate of the college when he graduated, and was a winner of the 2020 Chinese Academy of Sciences President Award.

Ajian Liu received the master's degree from the College of Information and Computer, Taiyuan University of Technology in 2018, and the Ph.D. degree with the Faculty of Information Technology, Macau University of Science and Technology in 2022. He is currently working as a Postdoctoral Fellow with the Institute of Automation, Chinese Academy of Sciences. His main research interests include deep learning and face anti-spoofing.

Jun Wan (Senior Member, IEEE) received the B.S. degree from the China University of Geosciences, Beijing, China, in 2008, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University, Beijing, in 2015. Since January 2015, he has been a Faculty Member with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, China, where he currently serves as an Associate Professor. His main research interests include computer vision and machine learning. He is an Associate Editor of the *IET Biometrics* from 2020 to 2022, the Area Chair of ICME 2021 and 2020, the Senior Program Committee for AAAI 2021, and has served as the Co-Editor for special issues in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.

Zhen Lei (Senior Member, IEEE) received the B.S. degree in automation from the University of Science and Technology of China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010, where he is currently a Professor. He has published more than 200 papers in international journals and conferences. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular. He is the winner of 2019 IAPR Young Biometrics Investigator Award. He served as the Area Chair for ECCV2022, ICPR2022, IJCB/ICB in 2014, 2015, 2016, and 2018 and FGR2015. He is an Associate Editor of *Pattern Recognition* journal.



Guodong Guo (Senior Member, IEEE) received the B.E. degree in automation from Tsinghua University, Beijing, China, the Ph.D. degree in computer science from the University of Wisconsin, Madison, WI, USA. He is the Head of the Institute of Deep Learning, Baidu Research, and also affiliated with the Department of Computer Science and Electrical Engineering, West Virginia University (WVU), USA. In the past, he studied, visited or worked in several places, including Institute of Automation, Chinese Academy of Sciences; INRIA, Sophia Antipolis, France; Ritsumeikan University, Kyoto, Japan; and Microsoft Research, Beijing, China; He authored a book, *Face, Expression, and Iris Recognition Using Learning-based Approaches* (2008), co-edited two books, *Support Vector Machines Applications* (2014) and *Mobile Biometrics* (2017), and coauthored a book, *Multi-Modal Face Presentation Attack Detection* (2020). He published over 200 technical papers, and he is the creator of the visaul body mass index estimator. His research interests include computer vision, biometrics, machine learning, and multimedia. He received the North Carolina State Award for Excellence in Innovation in 2008, a New Researcher of the Year from 2010 to 2011, and an Outstanding Researcher from 2017 to 2018, and from 2013 to 2014 at CEMR, WVU. He was selected the "People's Hero of the Week" by BSJB under Minority Media and Telecommunications Council in 2013. Two of his papers were selected as "The Best of FG'13" and "The Best of FG'15," respectively. He is an Associate Editor of several journals, including IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.