Cross-Batch Hard Example Mining With Pseudo Large Batch for ID vs. Spot Face Recognition

Zichang Tan, Ajian Liu, Jun Wan, *Senior Member, IEEE*, Hao Liu, Zhen Lei, *Senior Member, IEEE*, Guodong Guo, *Senior Member, IEEE*, and Stan Z. Li, *Fellow, IEEE*

***Abstract*— In our daily life, a large number of activities require identity verification, e.g., ePassport gates. Most of those verification systems recognize who you are by matching the ID document photo (ID face) to your live face image (spot face). The ID vs. Spot (IvS) face recognition is different from general face recognition where each dataset usually contains a small number of subjects and sufficient images for each subject. In IvS face recognition, the datasets usually contain massive class numbers (million or more) while each class only has two image samples (one ID face and one spot face), which makes it very challenging to train an effective model (e.g., excessive demand on GPU memory if conducting the classification on such massive classes, hardly capture the effective features for bisample data of each identity, etc.). To avoid the excessive demand on GPU memory, a two-stage training method is developed, where we first train the model on the dataset in general face recognition (e.g., MS-Celeb-1M) and then employ the metric learning losses (e.g., triplet and quadruplet losses) to learn the features on IvS data with million classes. To extract more effective features for IvS face recognition, we propose two novel algorithms to enhance the network by selecting harder samples for training. Firstly, a Cross-Batch Hard Example Mining (CB-HEM) is proposed to select the hard triplets from not only the current mini-batch but also past dozens of mini-batches (for convenience, we use batch to denote a mini-batch in the following), which can significantly expand the space of sample selection. Secondly, a Pseudo Large Batch (PLB) is proposed to virtually increase the batch size with a fixed GPU memory. The proposed PLB and CB-HEM can be employed simultaneously to train the network, which dramatically expands the selecting space by hundreds of times, where the very hard sample pairs especially the hard negative pairs can be selected for training to enhance the discriminative capability. Extensive comparative evaluations conducted on multiple IvS benchmarks demonstrate the effectiveness of the proposed method.**

***Index Terms*— Face recognition, ID vs. spot, deep learning, cross-batch hard example mining, pseudo large batch.**

Manuscript received July 7, 2021; revised October 21, 2021; accepted December 2, 2021. Date of publication April 12, 2022; date of current version April 26, 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 under Project 62106264, Project 61876179, and Project 61961160704; in part by the Key Project of the General Logistics Department under Grant AWS17J001; and in part by the Science and Technology Development Fund of Macau under Project 0008/2019/A1, Project 0025/2019/AKP, and Project 0070/2020/AMJ. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Soma Biswas. *(Corresponding author: Jun Wan.)*

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; guoguodong01@baidu.com).

Ajian Liu is with the Faculty of Innovation Engineering, Macau University of Science and Technology (MUST), Macau, China (e-mail: ajianliu92@gmail.com).

Jun Wan is with the Faculty of Innovation Engineering, Macau University of Science and Technology (MUST), Taipa, Macao, also with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100190, China (e-mail: jun.wan@nlpr.ia.ac.cn).

Hao Liu is with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 100190, China (e-mail: hao.liu2016@nlpr.ia.ac.cn).

Zhen Lei is with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), 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).

Stan Z. Li is with the Faculty of Innovation Engineering, Macau University of Science and Technology (MUST), Taipa, Macau, China, and also with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: stan.zq.li@westlake.edu.cn).

Digital Object Identifier 10.1109/TIP.2021.3137005

I. Introduction

FACE recognition [1]–[10] has been a thriving research field in the past decades, on account of its wide range of applications such as human identification [11]–[13], access control [14], face retrieval [15] and so on. In many real-world applications, face recognition is usually conducted by matching the live face (called the spot face) with the face in ID document (called ID face), which is called ID vs. Spot (IvS) face recognition [12]. IvS face recognition plays an important role in our daily lives, e.g., ePassport gates with face authentication system.

Until now, the IvS face recognition still confronts some critical challenges and needs profound studies. Different from general face recognition [4], [5], [8]–[10], [16], the datasets of IvS face recognition are usually captured from practical face authentication systems (e.g., ePassport gates and ID card gates), where a very large number of identities are usually accessible (up to millions and even tens of millions of identities) due to the high flow of people traffic but only two images (a live face photo and a ID face) for each identity can be captured. For such massive classes, if we directly train the network with softmax loss or its modifications (e.g., AM-softmax [8], [9]), numerous parameters are brought in the classifier layer, which poses a tremendous pressure on GPU resource. To avoid this, we take a two-stage training method for IvS face recognition. We first train the deep network on large datasets (e.g., MS-Celeb-1M [17]) for general face recognition with A-softmax loss function [5], which helps the

network to obtain a good initialization. Then, the network is finetuned on the IvS datasets with the metric learning loss (both triplet loss [18] and quadruplet loss [19] are employed), where the classifier layer can be removed in this stage. Triplet loss and quadruplet loss learn features from different perspectives, where triplet loss is constructed based on a triplet (an anchor, a positive and a negative sample) and quadruplet loss is constructed based on a quadruplet (four arbitrary samples with two of a positive pair and the other two of a negative pair). To our best knowledge, it is the first attempt to apply the quadruplet loss to the task of face recognition.

In metric learning, one critical challenge is how to select effective sample pairs in the training stage. Some previous works [20], [21] propose the idea of Batch Hard Example Mining (B-HEM) to select hard samples for training while ignoring easy samples. As shown in Fig. 1 (a), more discriminative features can be captured when training with B-HEM. However, only using the B-HEM is not sufficient. In B-HEM, the hard sample pairs can only be selected from the current batch with containing a limited number of images. For example, the maximal batch size can only be set to 384 with the 64-layer residual network on three 2080Ti GPUs (11G memory per GPU). In IvS face recognition, we find that a larger batch size leads to a larger selecting space and thus the better performance can be achieved as shown in Fig. 1 (b). To enlarge the selecting space, we propose two new algorithms. The first one is called Cross-Batch Hard Example Mining (CB-HEM). Although the network parameters are changing throughout the training process, the features of nearby iterations will not be changed so much. Thus, the features of past batches can also be an important reference when selecting hard sample pairs. In other words, the sample selection is not limited to the current batch but can be extended to dozens of past $M-1$ batches, where the selecting space can be extended by dozens of times. However, the number of crossed batches M cannot be set too large, where the network parameters would be changed a lot after so many iterations. Thus the second algorithm named Pseudo Large Batch (PLB) is proposed. It updates the network parameters every $PseudoN$ iterations by using accumulated gradients. In this way, the batch size can be virtually increased by $PseudoN$ times. Thus, when PLB is used with CB-HEM together, the searching space can be further expanded. Besides, we also propose a new loss term in PLB to build the connections among the samples in different iterations, where the samples in previous iterations can also be selected to calculate the loss in the current iteration.

The contributions of our work can be summarized as follows: (1) A novel method called Cross-Batch Hard Example Mining (CB-HEM) is proposed for IvS face recognition. Compared with previous B-HEM, it selects the hard sample pairs from both current and past batches rather than only the current batch, which helps capture more effective sample pairs for training. (2) A Pseudo Large Batch (PLB) method is proposed to virtually increase the batch size with breaking through the limitation of GPU memory. It can be used with CB-HEM concurrently to further expand the searching space, where more difficult sample pairs can be captured for training. (3) Both triplet and quadruplet losses are employed to optimize the network from different aspects. To our best knowledge, it is the first attempt to apply the quadruplet loss to IvS face recognition. (4) Extensive comparative evaluations demonstrate the superiority of the proposed method over several benchmark datasets including Private-IvS, Public-IvS and LFW-BLUFR.

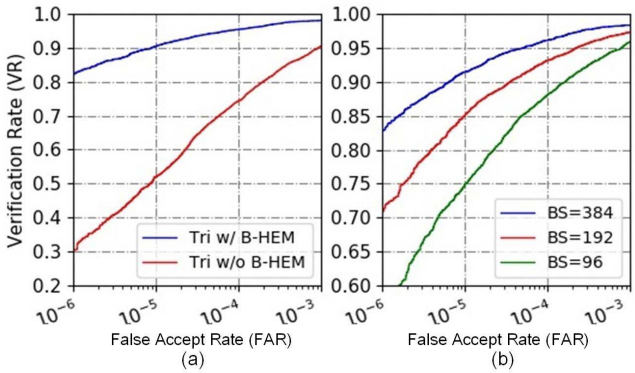


Fig. 1. The left: the comparisons of training the network with ('w/') or without ('w/o') Batch Hard Example Mining (B-HEM). The triplet loss (denoted by 'Tri') is employed as the loss function. The right: the comparisons of training the network with different batch size (denoted by 'BS'). The triplet loss with B-HEM is employed as the loss function. All networks are trained on Private-IvS-Train-S and evaluated on Private-IvS-Test (see Section IV-A for details).

II. Related Works

A. General Face Recognition

Face recognition [1]–[5], [8]–[10], [12], [13], [22], [23], which has been studied for more than 30 years, is a very classical problem in computer vision. In 1990s to early 2010s, the holistic approaches (e.g., linear subspace [24], manifold [25] and sparse representation [26]), local descriptors (e.g., Gabor [27] and LBP [28]) dominated the face recognition community. Most of those feature descriptors are handcrafted, which suffers from a lack of distinctiveness and compactness. Later, face recognition has achieved a series of breakthroughs owing to the great success of deep learning [29]–[39]. e.g., DeepFace [40], DeepID [3], DeepID2 [13] and FaceNet [18]. Very recently, many researchers find that the loss function plays an important role in face recognition, while traditional softmax loss is not powerful enough to extract the discriminative features. Thus, a series of loss functions [4], [5], [8]–[10], [41]–[47] are proposed to boost the performance, e.g., Center loss [41], L-softmax [4], A-softmax [5], AM-softmax [8], [9] and so on. All of those losses share the same idea to improve the discriminative capability: maximize the inter-class variations and minimize the intra-class variations. For example, center loss [41] is proposed to learn centers for each identity and minimize the intra-class distance by narrowing the distance among each sample and its corresponding center. L-softmax [4], A-softmax [5] and AM-softmax [8], [9] improve the feature discrimination by adding angular or cosine margin constraints. Moreover, some works adopt the attention mechanism [48], [49] or advanced architecture [50], and also achieve promising performance.

B. IvS Face Recognition

The studies about IvS face recognition are very rare although it faces so many challenges. To our best knowledge,

the first work about IvS face recognition is first studied by Starovoitov *et al.* [51], Starovoitov and Samal [52]. The authors employ the Hough Transform to localize the eyes. Then the whole face region is cropped and gradient maps are computed as feature maps for recognition. Recently, Shi and Jain [53], [54] and Zhu *et al.* [12] first apply the deep learning technology to this problem. Shi and Jain [53], [54] propose DocFace/DocFace+ with a Dynamic Weight Imprinting (DWI), which allows faster convergence and more generalizable representations. Zhu *et al.* [12] propose a dominant prototype softmax (DP-softmax), which makes deep learning applicable to large-scale classes. More recently, Albiero *et al.* [55] study the problem of IvS face recognition across age differences in adolescence. Our work is different from previous works. We propose two novel algorithms, namely CB-HEM and PLB, both of which help the network capture more difficult samples beyond the limitation of GPU memory.

C. Deep Metric Learning

Deep metric learning aims to learn a feature embedding space with large inter-variations while small intra-variations according to pairwise distances or similarities [18], [19], [56]–[61]. In those methods, contrastive loss [56] and triplet loss [18] are two classical deep metric learning methods. Contrastive loss learns a discriminative metric to narrow the distance of positive pairs and enlarge the distances of negative pairs to be large than a fixed threshold. For triplet loss, it encourages the features of a positive pair is farther than those of a negative pair (with respect to the same anchor) by a given margin. Extended from contrastive and triplet losses, quadruplet [19] and histogram loss [58] are also proposed in recent work. Recent pair-based losses aim to explore the full pair-wise relations between samples in a mini-batch. For example, N-pair loss [62] and Lifted Structure loss [57] are proposed to associate an anchor sample with a positive sample and multiple negative samples. Similar to triplet loss, those two losses [57], [62] learn to pull the positive to the anchor while pushing the negatives away from the anchor sample. Moreover, Ranked List loss [63] considers all positive and negative samples in a batch. Multi-Similarity (MS) loss [60] considers all pair in a batch and assigns a weight to each pair, which helps the network focus on useful pairs. Recently, proxy-based losses are proposed to improve the efficiency by employing some representative proxies for calculations. The classical losses including Proxy-NCA [64], Proxy-Anchor [65], SoftTriple [66]. For example, Proxy-Anchor assigns each class with a proxy, and then calculate the loss based on all pairs among all proxies and all samples. Sampling plays an important role in pair-based metric learning. Thus, hard mining [20], [67], [68] is proposed to improve training efficiency. Hermans *et al.* [20] propose a batch hard example mining in triplet loss, where only the hardest positive and negative pairs are selected for training, which improves the feature discrimination by learning the semantics from hard sample pairs. However, the batch hard example mining usually suffers from the GPU memory, where the hard sample pairs can only be selected from the current batch of a limited size.

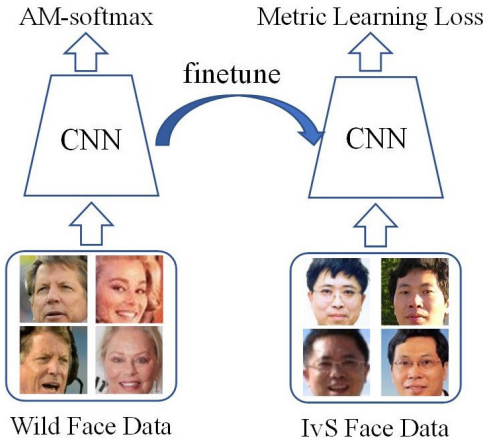


Fig. 2. An illustration of the two-step training manner for IvS face recognition.

III. The Proposed Approach

An illustration of the proposed method is shown in Fig. 2, where a two-stage training manner is employed to learn an effective model for IvS face recognition. More specifically, the network is first pretrained on MS-Celeb-1M [17] of general face recognition with AM-softmax loss function [8], [9]. Then, a metric learning way is employed to finetune the network on the IvS dataset. In the second stage, both triplet and quadruplet losses are employed as the loss function. Besides, two novel algorithms namely CB-HEM and PLB are proposed to select more difficult sample pairs for training.

A. Triplet Loss

Triplet loss [18] is learned on a series of triplets $\{\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n\}$, where $\mathbf{x}_a$, $\mathbf{x}_p$ and $\mathbf{x}_n$ indicate the feature vectors of anchor, positive and negative samples, respectively. Mathematically, the triplet loss can be represented as:

$$L_{tri} = \sum_{\substack{a,p,n \\ y_a=y_p\neq y_n}} [d(\mathbf{x}_a, \mathbf{x}_p) - d(\mathbf{x}_a, \mathbf{x}_n) + m_1]_+ \quad (1)$$

where y_i indicates the label of the feature vector $\mathbf{x}_i$, and $[z]_+ = \max(z, 0)$. m_1 represents the margin hyperparameter to control the differences of intra- and inter- distances. In our experiments, m_1 is set to 0.2 by our experience. $d(\mathbf{r}_1, \mathbf{r}_2)$ indicates a metric function to measure the distance between $\mathbf{r}_1$ and $\mathbf{r}_2$ and we set $d(\mathbf{r}_1, \mathbf{r}_2) = \|\mathbf{r}_1 - \mathbf{r}_2\|_2$.

B. Quadruplet Loss

Different from triplet loss, quadruplet loss learns features according to the relative distance between arbitrary positive and negative pairs. We define a quadruplet as $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l, \mathbf{x}_k\}$, where $\mathbf{x}_i, \mathbf{x}_j$ come from the same identity and $\mathbf{x}_l, \mathbf{x}_k$ from any two different identities. The quadruplet loss is represented as:

$$L_{qua} = \sum_{\substack{i,j,l,k \\ y_i=y_j, y_l\neq y_k}} [d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_l, \mathbf{x}_k) + m_2]_+ \quad (2)$$

where m_2 indicates the margin hyperparameter and it is set to 0.2. As shown in the above equation, the quadruplet loss aims to make the distances between any two positive samples less than that of any two negative samples.

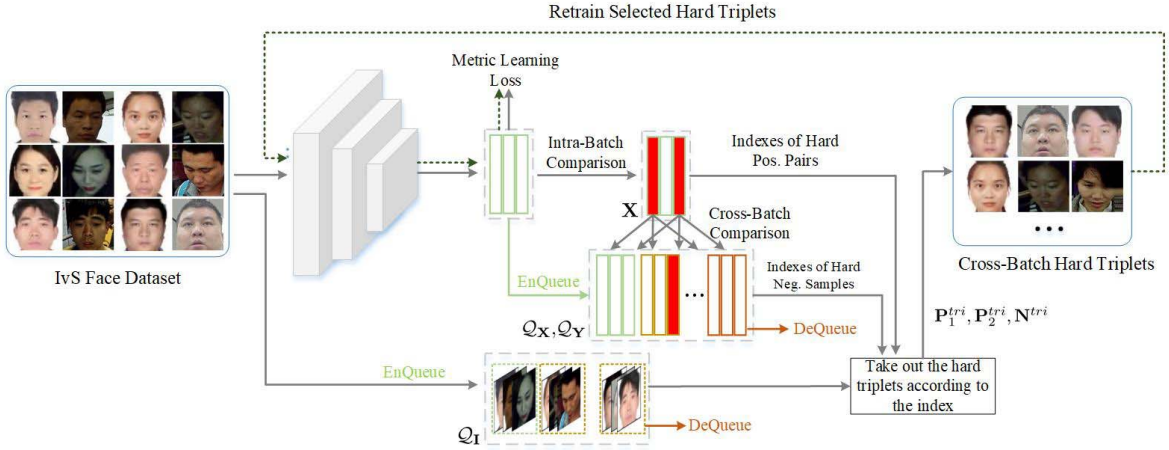


Fig. 3. An illustration of the proposed Cross-Batch Hard Example Mining (CB-HEM). Our CB-HEM can generate cross-batch hard triplets to train the network more effectively. We first conduct an intra-batch comparison to select the hard positive pairs from the current batch. Then, we compare them with a feature queue containing the features from past batches to select hard negative samples. Those selected hard positive pairs and negative samples form cross-batch hard triplets, which will be retrained to improve the network's discriminative capability.

C. Batch Hard Example Mining

The triplet loss with B-HEM [20] is an improved version of the original semi-hard triplet loss [18]. Assume each batch contains N images and a certain number of identities with several images per identity. Then, each image would be set as the anchor sample and the hardest positive and negative samples with respect to each anchor would be selected within a batch when formulating the triplets for training. The triplet loss with B-HEM can be formulated as:

$$L_{tri}^{bh} = \sum_{a=1}^{N} [\overbrace{\max_{\substack{p=1,\cdots,N \\ y_a=y_p}} d(\mathbf{x}_a, \mathbf{x}_p)}^{hardest\ positive} - \overbrace{\min_{\substack{n=1,\cdots,N \\ y_a\neq y_n}} d(\mathbf{x}_a, \mathbf{x}_n)}^{hardest\ negative} + m_1]_+ \tag{3}$$

Inspired by the above formula, we also can formulate a hard example mining for the quadruplet loss. We first traverse all positive and negative pairs, and select the most difficult K positive and K negative pairs. Then those selected pairs are randomly paired to form hard quadruplets for computing the quadruplet loss. In our experiments, K is set to be equal to the batch size N. The quadruplet loss with B-HEM can be written as:

$$L_{qua}^{bh} = \sum_{\substack{(i,j)\in \mathbf{P}_{pair}^{hard\,K} \\ (l,k)\in \mathbf{N}_{pair}^{hard\,K}}} [\overbrace{d(\mathbf{x}_i, \mathbf{x}_j)}^{hard\,K\ positive} - \overbrace{d(\mathbf{x}_l, \mathbf{x}_k)}^{hard\,K\ negative} + m_2]_+ \tag{4}$$

where $\mathbf{P}_{pair}^{hard\,K}$ and $\mathbf{N}_{pair}^{hard\,K}$ indicate the selected hard positive and negative pairs.

D. Cross-Batch Hard Example Mining

In B-HEM, the hard sample pairs are only selected from the current batch while the samples of past batches are ignored. Although the network is updated at each iteration, but the features of nearby iterations won't be changed so much. Thus, the samples in the past batches also can be important references when selecting the hard sample pairs in the current batch. Inspired by this, we propose a CB-HEM, which extends the selecting space to nearby M batches (a current batch and $M-1$ past batches). An illustration of our proposed CB-HEM is shown in Fig. 3 and the details of the algorithm is shown in Algorithm 1. Given a batch of images $\mathbf{I} = [\mathbf{I}_1, \cdots \mathbf{I}_N]$ and corresponding labels $\mathbf{Y} = [y_1, \cdots, y_N]$, the corresponding batch features $\mathbf{X} = [\mathbf{x}_1, \cdots \mathbf{x}_N]$ are extracted by the network $\mathcal{F}(\cdot; \Theta)$, where Θ denotes the network parameters needed to be learned. Based on the batch features $\mathbf{X}$ and labels $\mathbf{Y}$, we first capture the batch hard triplets and quadruplets according to Eq. (3) and Eq. (4), respectively. Then, the network is first updated by using B-HEM according to the summed loss as following:

$$L^{bh} = L_{tri}^{bh} + L_{qua}^{bh} \tag{5}$$

Subsequently, we detach $\mathbf{X}$ from the computational graph and denote them as $\tilde{\mathbf{X}}$, which becomes just numerical vectors of taking up a small amount of GPU memory. Then, we collect detached features $\tilde{\mathbf{X}}$, images $\mathbf{I}$ and labels $\mathbf{Y}$ by using three queues, namely $\mathcal{Q}_{\mathbf{X}}$, $\mathcal{Q}_{\mathbf{I}}$ and $\mathcal{Q}_{\mathbf{Y}}$, respectively.

Later, we aim to capture the cross-batch hard triplets according to $\tilde{\mathbf{X}}$ and $\mathcal{Q}_{\mathbf{X}}$. Note each identity only contains two images, and these two images would be placed into the same batch. In other words, the selection of positive sample pairs can be only considered in the current batch. Thus, we first select the hard positive sample pairs from the current batch according to the pair-wise distance matrix $\mathbf{D}_1 \in R^{N\times N}$ across all within-batch samples. According to the idea of hard example mining, only a certain small proportion (denoted by r) of the most difficult positive pairs will be selected out and their corresponding indexes are denoted as $\mathbf{P}_1^{tri}$ and $\mathbf{P}_2^{tri}$. For convenience, we denote this process as $SelHardPos$. Later, a cross-batch comparison is employed to select hard negative samples. First, a distance matrix $\mathbf{D}_2 \in R^{N\times MN}$ among the features of the current batch and past $M-1$ batches will be calculated. Based on $\mathbf{D}_2$, the hardest negative sample for each positive sample (with minimum distance) would be found out. More specifically, for two positive samples (one pair) in $\mathbf{P}_1^{tri}$ and $\mathbf{P}_2^{tri}$, all of them would find their hardest negative samples but only the most difficult one will be retained and

Algorithm 1 Cross-Batch Hard Example Mining

Input: Training dataset $\mathcal{D}=\{\mathbf{I}_i, y_i\}$; Feature extraction network $\mathcal{F}(\cdot;\Theta)$; Learning rate η; Number of crossed batches M; The proportion r of the selected hard pairs.
Output: Feature extraction network $\mathcal{F}(\cdot;\Theta)$.

Initialize network $\mathcal{F}(\cdot;\Theta)$;
Initialize queues $\mathcal{Q}_{\mathbf{I}}, \mathcal{Q}_{\mathbf{Y}}, \mathcal{Q}_{\mathbf{X}}, \mathcal{Q}^{tri}$ as empty ;
while *not convergence* **do**
  //// training with batch hard example mining
  Sampling data $\mathbf{I}=[\mathbf{I}_1,\cdots,\mathbf{I}_N], \mathbf{Y}=[y_1,\cdots,y_N]$;
  Extract batch features $\mathbf{X}=\mathcal{F}(\mathbf{I};\Theta)$;
  Calculate L^{bh} according to Eq. 5;
  Update $\Theta \leftarrow \Theta - \eta\frac{\partial L_{bhm}}{\partial \Theta}$ with SGD;
  // get into the queues
  Detach features $\tilde{\mathbf{X}}=detach(\mathbf{X})$;
  $EnQueue(\mathcal{Q}_{\mathbf{I}},\mathbf{I})$,$EnQueue(\mathcal{Q}_{\mathbf{Y}},\mathbf{Y})$,$EnQueue(\mathcal{Q}_{\mathbf{X}},\tilde{\mathbf{X}})$
  //// select cross-batch hard triplets
  // intra-batch comparisons
  Calculate the distance $\mathbf{D}_1=CalDist(\mathbf{X})$;
  $\mathbf{P}_1^{tri},\mathbf{P}_2^{tri}=SelHardPos(\mathbf{D}_1,\mathbf{Y},\mathcal{Q}_{\mathbf{Y}},\mathcal{Q}_{\mathbf{I}},r)$;
  // cross-batch comparisons
  Calculate the distance $\mathbf{D}_2=CalDist(\mathbf{X},\mathcal{Q}_{\mathbf{X}})$;
  $\mathbf{N}^{tri}=SelHardNeg(\mathbf{D}_2,\mathbf{Y},\mathcal{Q}_{\mathbf{Y}},\mathcal{Q}_{\mathbf{I}},\mathbf{P}_1^{tri},\mathbf{P}_2^{tri})$;
  $EnQueue(\mathcal{Q}^{tri},(\mathbf{P}_1^{tri},\mathbf{P}_2^{tri},\mathbf{N}^{tri}))$;
  // out of queues
  if *QueueLength*$(\mathcal{Q}_{\mathbf{I}})>=M$ **then**
    $DeQueue(\mathcal{Q}_{\mathbf{I}})$,$DeQueue(\mathcal{Q}_{\mathbf{Y}})$,$DeQueue(\mathcal{Q}_{\mathbf{X}})$
  end
  ////retrain the selected cross-batch triplets
  if *QueueLength*$(\mathcal{Q}^{tri})>N$ **then**
    $\mathbf{P}_1^{tri},\mathbf{P}_2^{tri},\mathbf{N}^{tri}=DeQueue(\mathcal{Q}^{tri})$;
    Extract features for $\mathbf{P}_1^{tri},\mathbf{P}_2^{tri},\mathbf{N}^{tri}$;
    Calculate L_{tri} according to Eq. (1);
    Update $\Theta \leftarrow \Theta - \eta\frac{\partial L_{tri}}{\partial \Theta}$ with SGD;
  end
end

the corresponding positive sample will be set as the anchor sample. This process is represented as $SelHardNeg$ for short, and the indexes of selected hard negative samples are denoted as $\mathbf{N}^{tri}$. Those cross-batch hard triplets (i.e., $\mathbf{P}_1^{tri}$, $\mathbf{P}_2^{tri}$ and $\mathbf{N}^{tri}$) are then input to the queue $\mathcal{Q}^{tri}$. When the number of samples in $\mathcal{Q}^{tri}$ reaches a certain number (e.g., the batch size N), we will take the raw images of those hard triplets out, and input them into the network for training again, where the network is updated according to Eq. (1).

E. Pseudo Large Batch

As shown in Fig. 1 (b), the better performance can be achieved when training with a large batch size, where the harder samples can be discovered. However, limited by GPU resource, the batch size cannot be set too large. Therefore, we propose a PLB algorithm as illustrated in Algorithm 2 to virtually increase the training batch size. The core idea in PLB is to update every $PseudoN$ iterations with using the accumulated gradients of all those iterations, which virtually increases the batch size by $PseudoN$ times. Of course, simply using the accumulated gradients for updating only virtually increases the batch size, but cannot explore the correlations

Algorithm 2 Training With Pseudo Large Batch

Input: Training dataset $\mathcal{D}=\{\mathbf{I}_i, y_i\}$; Feature extraction network $\mathcal{F}(\cdot;\Theta)$; Learning rate η; Pseudo expansion times $PseudoN$.
Output: Feature extraction network $\mathcal{F}(\cdot;\Theta)$.

Initialize network $\mathcal{F}(\cdot;\Theta)$;
while *not convergence* **do**
  $\nabla\Theta = 0$, $\tilde{\mathbf{X}}_{plb}=[\,]$;
  for $iter=1; iter \le PseudoN$ **do**
    Sampling data $\mathbf{I}=[\mathbf{I}_1,\cdots,\mathbf{I}_N], \mathbf{Y}=[y_1,\cdots,y_N]$;
    Extract batch features $\mathbf{X}=\mathcal{F}(\mathbf{I};\Theta)$;
    $\tilde{\mathbf{X}}=detach(\mathbf{X})$, $\tilde{\mathbf{X}}_{plb}=[\tilde{\mathbf{X}}_{plb},\tilde{\mathbf{X}}]$;
    Calculate L_{plb} according to Eq. 7;
    $\nabla\Theta \leftarrow \nabla\Theta + \frac{1}{PseudoN}\cdot\frac{\partial L_{plb}}{\partial\Theta}$;
  end
  Update $\Theta \leftarrow \Theta - \eta\nabla\Theta$ with SGD;
end

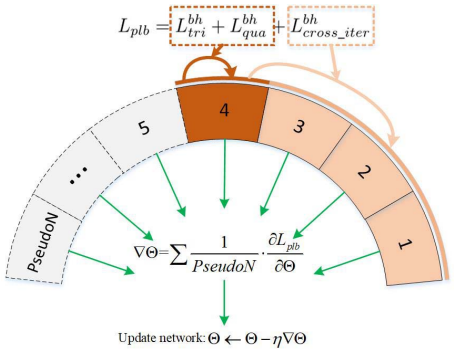


Fig. 4. An illustration of the proposed Pseudo Large Batch (PLB).

between different iterations. To achieve this, a loss $L_{cross\_iter}^{bh}$ is constructed to help the network select cross-iteration hard examples. Mathematically, $L_{cross\_iter}^{bh}$ can be represented as follows:

$$L_{cross\_iter}^{bh}=\sum_{a=1}^{N}[\overbrace{\max_{\substack{p=1,\cdots,N\\ y_a=y_p}} d(\mathbf{x}_a,\mathbf{x}_p)}^{hardest\ positive}-\overbrace{\min_{\substack{\tilde{\mathbf{x}}_n\in\tilde{\mathbf{X}}_{plb}\\ y_a\neq y_n}} d(\mathbf{x}_a,\tilde{\mathbf{x}}_n)}^{hardest\ negitive}+m_1] \tag{6}$$

where $\tilde{\mathbf{X}}_{plb}$ denotes the detached features collected from previous iterations (see Algorithm 2 for details). In this loss function, the anchor and positive samples are chosen from the current iteration (each identity only contains two samples and all of them would be placed in the same iteration), while the negative samples can be selected from any previous iteration. The training loss for each iteration can be written as:

$$L_{plb}=L_{tri}^{bh}+L_{qua}^{bh}+L_{cross\_iter}^{bh} \tag{7}$$

Obviously, the term $L_{cross\_iter}^{bh}$ captures hard sample pairs across iterations, which helps to improve the network's discriminative capability. Besides, the traditional batch in CB-HEM can be replaced with our proposed PLB, where the network is also updated every $PseudoN$ iterations with the accumulated gradients and the loss function of L^{bh} is also replaced with L_{plb} (the part of finding cross-batch hard triplets

Fig. 5. Some examples of Private-IvS dataset. Each identity contains two face images, namely one ID face and one spot face. The ID face is usually in low resolution due to the image compression, while the spot face is captured in the unconstrained environment with large variations of illuminations, poses, background and so on.

and retraining will not be changed). In this way, the proposed CB-HEM and PLB can be employed simultaneously to train the network, where the selecting space will be dramatically expanded by $M \times PseudoN$ times.

F. Algorithm Acceleration

To improve the efficiency of the proposed method, we conduct the algorithm acceleration from the following aspects:

- *Find the hard positive samples based on small distance matrixes (denoted as A).* Each identity contains only two samples and all of those two samples will be placed in a same batch. Thus, for the PLB, the hard positive samples can be selected according to $PseudoN$ small distance matrixes, each of which contains all pair-wise distances of the whole batch. Intuitively, find the hardest positive pairs in a large distance matrix (e.g., a pseudo large batch with the size of $PseudoN \cdot N \times PseudoN \cdot N$) would be obviously tough than finding those in a small distance matrix (e.g., a small batch with the size of $N \times N$).
- *Simplifying distance calculation (denoted as B).* The distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is calculated as:

$$D_{ij}^2 = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 = \|\mathbf{x}_i\|^2 - 2{\mathbf{x}_i}^T\mathbf{x}_j + \left\|\mathbf{x}_j\right\|^2 \quad (8)$$

  Note that the features are normalized by L2 normalization, where $\|\mathbf{x}_i\|^2 = \left\|\mathbf{x}_j\right\|^2 = 1$. Therefore, the distance can be calculated by $D_{ij}^2 = 2 - 2{\mathbf{x}_i}^T\mathbf{x}_j$, where the calculations of $\|\mathbf{x}_i\|^2$ and $\left\|\mathbf{x}_j\right\|^2$ can be omitted.

IV. Experiments

A. Datasets

In our experiments, all networks are first pretrained on MS-Celeb-1M dataset [17], and then finetuned on the Private-IvS dataset (a private dataset of IvS face recognition) or Megaface-bisample [69]. Then, we evaluate the proposed method on three datasets, including the test part of Private-IvS, Public-IvS [12] and LFW-BLUFR [70], [71]. We will introduce those datasets in the following.

1) MS-Celeb-1M: MS-Celeb-1M is the largest wild dataset with containing 10 million images of 98,685 celebrities. All those images are crawled from the IMDB website[1] and the dataset contains much noise. Thus, we use a cleaned version of this dataset for training according to the list [72], where only 5 million images of 79,077 identities have remained.

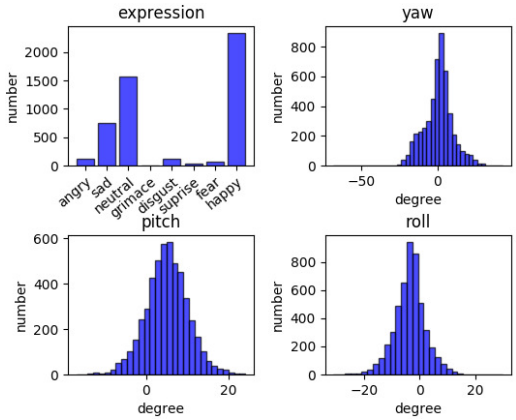


Fig. 6. The distributions of facial expressions and head poses in Public-IvS dataset. There are eight types of facial expressions, including angry, sad, neutral, grimace, disgust, surprised, fear and happy. Moreover, we use the angles of yaw, pitch and roll to denote a head pose, where different combinations of yaw, pitch and roll indicate different head poses. The distributions show large variations in facial expressions and head poses.

2) Private-IvS: Private-IvS is the dataset for IvS face recognition, and each identity in this dataset contains two face images (one ID face and one spot face). The ID face is captured with frontal face, clean background, neutral expression and so on, while the spot face is captured by the on-site devices (e.g., ID card gates), with large variations in background, head pose, expression, illumination and so on. In our experiments, we divide this dataset into three subsets: Private-IvS-Train-L(arge), Private-IvS-Train-S(mall) and Private-IvS-Test, which contains 2 million, 500,000 and 10,000 identities, respectively. The Private-IvS-Train-S is a subset of Private-IvS-Train-L, but the training sets and the test set are non-overlapped. When evaluating the proposed method, all the images in the test set are paired, where 10,000 positive pairs and about 400 million negative pairs are generated for testing.

3) Public-IvS: Public-IvS is a public evaluating dataset for IvS face recognition. All people in this dataset are public characters, such as politicians, teachers and researchers and so on. The images are crawled from the internet, like BaiduBake[2] or official pages. After crawling, those images are manually cleaned by students and staff. We use Baidu Face API[3] to analyze the distributions of emotions and head poses on this dataset as shown in Fig. 6. Finally, this dataset contains 1,262 identities and 5,507 images, and all of those images would be paired together during the testing stage.

4) Megaface-Bisample & LFW-BLUFR: We also evaluate the proposed method on the open MF2 dataset [69] following the Megaface-bisample protocol [12]. MF2 contains 657,559 identities but only two samples are randomly selected for each identity to mimic the bisample data. In the testing stage, the model is evaluated following the BLUFR protocol [71] on LFW [70]. More details about Megaface-bisample and LFW-BLUFR can be founded in the work [12].

[1] https://www.imdb.com/

[2] http://baike.baidu.com/

[3] https://ai.baidu.com/tech/face/detect

TABLE I

THE ANALYSIS OF M ON PRIVATE-IVS-TEST DATASET. 'N/A' DENOTES THE NETWORK IS ONLY TRAINED BY B-HEM OF TRIPLET AND QUADRUPLET LOSSES. THE TOP-2 RESULTS ARE HIGHLIGHTED

M	Verification Rate (VR)		
	FAR=1e-6	FAR=1e-5	FAR=1e-4
N/A	82.00	90.40	95.38
2	84.56	92.72	96.76
5	86.18	92.92	96.76
10	86.00	93.24	97.02
20	86.36	**93.64**	96.98
40	**86.92**	**93.62**	**97.22**
80	**87.38**	93.62	**97.00**

TABLE II

THE ANALYSIS OF RATIO r ON PRIVATE-IVS-TEST DATASET. 'N/A' DENOTES THE NETWORK IS ONLY TRAINED BY B-HEM OF TRIPLET AND QUADRUPLET LOSSES. THE TOP-2 RESULTS ARE HIGHLIGHTED

r	Verification Rate (VR)		
	FAR=1e-6	FAR=1e-5	FAR=1e-4
N/A	82.00	90.40	95.38
0.05	84.50	93.24	96.80
0.1	86.32	93.38	97.00
0.2	85.90	**93.74**	**97.22**
0.4	**86.92**	**93.62**	**97.18**
0.6	**86.68**	93.30	97.02
0.8	86.54	93.24	96.80
1.0	85.88	93.14	96.62
1.2	85.10	91.94	96.34
1.4	85.00	91.66	96.18

B. Settings and Metrics

All face images are detected by FaceBox [73] and then aligned the face by five landmarks (including two eyes, nose tip and two mouth corners). Then all faces are cropped and resized to the size of 120×120. In the training stage, the images are augmented by random flip. We conduct all experiments by using Pytorch, and a Stochastic Gradient Descent (SGD) with a weight decay of 0.0005 and a momentum of 0.9 is adopted to optimize the network. Following the work [5], [12], we adopt a 64-layer residual network is adopted as our backbone. In the first training stage, the network is trained by using AM-softmax [8], [9] on MS-Celeb-1M. Then, the network is finetuned on our Private-IvS dataset (e.g., Private-IvS-Train-L or Private-IvS-Train-S). In both two stages, 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 the batch size of 384.

In the evaluation stage, both the features of the raw image and its flipped copy would be extracted and then concatenated together as the final face feature. For any two face images, the score is obtained by calculating the cosine distance between their corresponding features. The ROC curve is employed as the evaluating metric. The verification rate (VR) at low false acceptance rate (FAR) important reference criteria especially in real application since false acceptance gives higher risks than false rejection.

C. Parameter Analysis

In this section, we mainly investigate the effects of some parameters on performance, including the number of crossed batches M, the proportion r of the selected hard pairs in CB-HEM and the pseudo expansion time $PseudoN$ in PLB. In this section, all networks are trained with Private-IvS-Train-S and evaluated with Private-IvS-Test.

1) The Number of Crossed Batches M in CB-HEM: M indicates the number of crossed batches that can be employed to select cross-batch hard triplets. We conduct the experiments with various M to search its optimal value (the proportion r of the selected hard pairs is set to 0.4). The network only trained by B-HEM of triplet and quadruplet losses is labeled as 'N/A' is also taken for comparisons, which can clearly show the performance improvement of each setting.

As shown in Table I, the optimal performance is hardly achieved when M is set to too small or too large. When M is set to too small (e.g., 2), the space of sample selection is only expanded a few times. At this time, the performance improvement is also relatively small, and there is still potential to pick harder cross-batch triplets when employing a larger M. When M is set to too large, the network has been updated many times and the features of past early batches may be out-of-date, which is less helpful for the network to select harder sample pairs. As shown in Table I, when M is set to be larger than 40, it is hard to further improve the performance while the computations will increase a lot. Thus, $M = 40$ is an optimal value with achieving the highest verification rates at FAR=1e-5 and FAR=1e-6. At the same time, the computations when setting M to 40 will not increase too much. M will be set to 40 in the following experiments.

2) The Proportion r of the Selected Hard Pairs in CB-HEM: r indicates the proportion of the number of the selected hard positive pairs and the size of a batch. Lots of experiments with varying r from 0.05 to 1.4 are conducted to find an optimal value. Note that $r > 1$ indicates an oversampling on the hardest positive sample pairs (also the cross-batch hard triplets) is employed. For example, $r = 1.4$ indicates 40% of the most difficult positive sample pairs are repeated (we select negative samples for those pairs in a same way). As shown in Table II, it achieves the best performance at FAR=1e-5 and FAR=1e-6 when setting r to 0.2. Although the higher performance can be achieved at FAR =1e-4 when $r = 0.4$ is employed, lots of computations are brought (twice as many triplets will be retrained). To balance the accuracy and efficiency, we adopt $r = 0.2$ in the following experiments. Moreover, 'N/A' indicates the model trained only with B-HEM of triplet and quadruplet losses but without CB-HEM. Take 'N/A' for comparisons helps us to know the improvement of each setting in CB-HEM. When r is too small, the learning of hard examples will be far from enough. When r is large, lots of easy pairs would be selected, which is useless for network training. What's more, experimental results show that employing an oversampling hardly improves the performance. This may be because the repeated samples contain much redundant information.

3) The Expansion Times $PseudoN$ in PLB: In PLB, we have virtually increased the batch size by $PseudoN$ times. Generally speaking, large $PseudoN$ means the network can select more difficult cross-batch triplets in a large set. At the

TABLE III

THE ANALYSIS OF *PseudoN* ON PRIVATE-IVS-TEST DATASET. 'N/A' DENOTES THE NETWORK IS ONLY TRAINED BY THE PROPOSED CB-HEM. THE TOP-2 RESULTS ARE HIGHLIGHTED

PseudoN	Verification Rate (VR)		
	FAR=1e-6	FAR=1e-5	FAR=1e-4
N/A	85.90	93.74	97.22
2	**88.70**	94.82	**97.88**
5	**89.18**	**95.40**	**98.06**
10	88.40	**95.24**	**97.88**
15	87.86	94.78	97.82

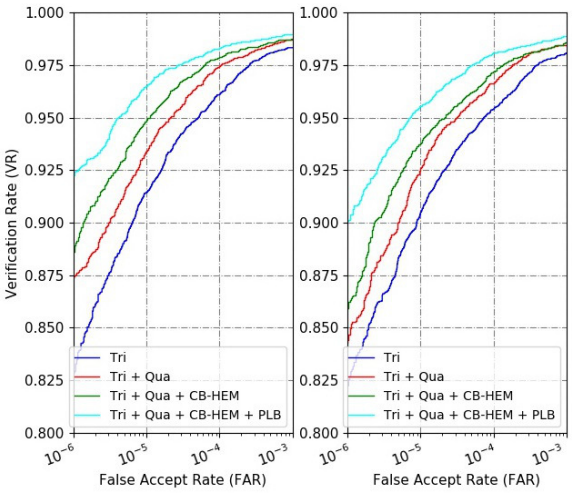


Fig. 7. Ablation studies on Private-IvS-Test dataset. The results shown in the left and right figures are trained on Private-IvS-Train-L and Private-IvS-Train-S, respectively. 'Tri', 'Qua', 'CB-HEM' and 'PLB' indicate triplet loss, quadruplet loss, CB-HEM and PLB, respectively.

same time, the number of the network updates will be reduced by $PseudoN$ times (the updates of retraining samples are not included) and the training efficiency has dropped. Thus, large $PseudoN$ does not necessarily get good performance although it helps to select more difficult sample pairs. To select an optimal value for $PseudoN$, a series of experiments are conducted as shown in Table III. The highest performance is achieved when setting $PseudoN$ to 5, where the accuracies at FAR=1e-4, FAR=1e-5 and FAR =1e-6 reach to 98.06%, 95.40% and 89.18%. In the following experiments, the value of $PseudoN$ is set to 5.

D. Ablation Studies

The ablation studies are conducted with the employed components, including Quadruplet loss (Qua), CB-HEM and PLB. The experiments are conducted on Private-IvS dataset, where both Private-IvS-Train-L and Private-IvS-Train-S are employed for training, and Private-IvS-Test and Public-IvS are employed for testing. The baseline model is trained with the triplet loss and B-HEM, then we gradually add the above components to the baseline model.

As shown in Fig. 7 and Fig. 8, all the above components can improve the performance on both Private-IvS-Test and Public-IvS datasets whether training on Private-IvS-Train-L (a large set with millions of classes) or Private-IvS-Train-S (a small set with hundreds of thousands of classes). For example, when training on Private-IvS-Train-L, quadruplet loss, CB-HEM and PLB improve the performance by about

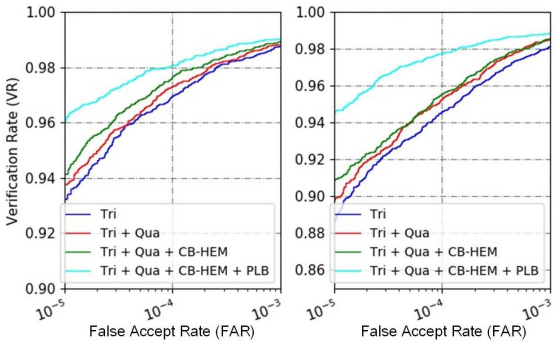


Fig. 8. Ablation studies on Public-IvS dataset. The results shown in the left and right figures are trained on Private-IvS-Train-L and Private-IvS-Train-S, respectively. 'Tri', 'Qua', 'CB-HEM' and 'PLB' indicate triplet loss, quadruplet loss, CB-HEM and PLB, respectively.

TABLE IV

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

Method	Verification Rate (VR)		
	FAR=1e-6	FAR=1e-5	FAR=1e-4
AM-softmax‡	45.46	62.10	78.28
Angular* [59]	82.36	90.92	95.82
Lifted* [57]	82.76	91.92	96.38
MS Loss* [60]	78.80	89.16	95.10
N-pair* [62]	77.38	87.84	94.84
Tri + B-HEM*	82.00	90.40	95.38
Tri + B-HEM†	82.82	91.42	96.08
Ours*	**89.18**	**95.48**	**98.06**
Ours†	**92.10**	**96.50**	**98.30**

‡ only pretrained on MS-Celeb-1M; finetuned on * Private-IvS-Train-S and † Private-IvS-Train-L.

4.5%, 1.1% and 1.5% at FAR=1e-6, respectively. When training on Private-IvS-Train-S, the corresponding performance is improved by 1.9%, 1.5% and 3.1%, respectively. Those performance improvements verify the effectiveness of proposed components, and also show that training with very hard sample pairs helps to improve the model's discriminative capability. Note that the Public-IvS dataset is collected from the web. It still contains some noises although it has been manually cleaned, which makes the performance improvement very difficult.

E. Comparisons to Prior Arts

1) Results on Private-IvS Dataset: For Private-IvS dataset, the baseline method, namely training the network with Triplet loss and B-HEM (denoted as Tri + B-HEM), is also employed for comparisons. Moreover, we also place the performance of our pretrained model (only pretrained on MS-Celeb-1M with AM-softmax) to show the benefits of finetuning on IvS dataset. Besides, we also implement four classical methods in metric learning, namely Angular [59], Lifted [57], MS Loss [60] and N-pair [62], for comparisons. The results are shown in Table IV. The pretrained model of AM-softmax can only achieve a very low performance on Private-IvS-Test dataset. More specifically, the accuracy at FAR=1e-6 is lower about 40% compared with our baseline method. This also shows that finetuning on IvS dataset is very necessary to bridge the gap between the datasets of IvS face recognition and general face recognition. Compared with the baseline method, the performance can be improved by 9.28% and 9.96% when training

TABLE V

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

Method	Verification Rate (VR)		
	FAR=1e-5	FAR=1e-4	FAR=1e-3
Angular* [59]	93.00	96.82	98.71
Lifted* [57]	89.48	95.28	98.42
MS Loss* [60]	90.43	95.77	98.38
N-pair* [62]	87.82	94.09	97.66
COTS-1 [54]	83.78	89.92	92.90
COTS-2 [54]	**94.74**	97.03	97.88
CenterFace [41]	35.97	53.30	69.18
SphereFace [5]	53.21	69.25	83.11
DocFace+ [54]	91.88	96.48	98.40
LBL(DPS) [12]	93.62	97.21	98.83
Tri + B-HEM*	88.52	94.56	98.13
Tri + B-HEM†	93.16	96.93	98.75
Ours*	94.48	**97.72**	**98.81**
Ours†	**95.95**	**98.03**	**99.01**

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

TABLE VI

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

Method	Verification Rate (VR)		
	FAR=1e-5	FAR=1e-4	FAR=1e-3
Lifted [57]	53.45	75.46	90.50
N-pairs [62]	50.30	73.40	90.16
LBL(DPS) [12]	73.86	88.03	95.68
Tri + B-HEM	61.93	85.40	94.43
Ours	**74.86**	**92.24**	**96.87**

on Private-IvS-Train-L and Private-IvS-Train-S, respectively. The considerable improvements show the proposed method can markedly enhance the model's discriminative capability.

2) Results on Public-IvS Dataset: For Public-IvS dataset, we compare the proposed method with previous state-of-the-art methods as shown in Table V. In addition to academic methods, two Commercial-Off-The-Shelf (COTS) face matchers, namely COTS-1 and COTS-2, are also employed for comparisons. The proposed method achieves outperforms all previous methods. For example, the proposed method trained on Private-IvS-Train-L outperforms LBL(DPS), DocFace+ and COTS-2 by 2.33%, 4.07% and 1.21% at FAR=1e-5, respectively. When training on Private-IvS-Train-S (only 1 million images of 500 thousand identities are included), our method also can achieve a very high accuracy of 94.48%, which is higher than most previous methods, e.g., LBL(DPS) and DocFace+. Note that LBL(DPS) is trained with about 4 million images of 2 million identities, which are much more than our Private-IvS-Train-S'.

3) Results on LFW-BLUFR Dataset: For LFW-BLUFR dataset, LBL(DPS) [12], Lifted Struct [57], N-pairs [62] and our baseline method Tri + B-HEM are taken for comparisons. For N-pairs and Lifted Struct, the results in the work [12] are reported. Our approach performs best with achieving the verification rate of 74.86%, 92.24% and 96.87% at FAR=1e-5, FAR=1e-4 and FAR=1e-3, respectively. The proposed method improves the previous best accuracy by about 1.0% at FAR=1e-5, which shows the proposed method also can perform well on bisample face dataset in the wild.

F. Discussions

1) Algorithm Acceleration: The experimental results of algorithm acceleration are shown in Fig 9. The modification

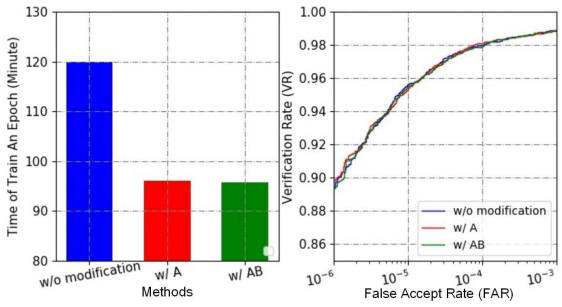


Fig. 9. Experimental results of the proposed algorithm acceleration. In both figures, 'w/o modification', 'w/ A' and 'w/ AB' indicate conducting the experiment without A and B, with A and with A and B, respectively. The left and right figures show comparisons of time consumption and performance, respectively.

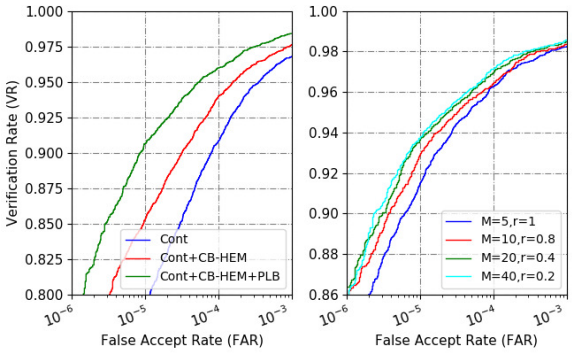


Fig. 10. (a) Experimental results of using Contrastive loss; (b) Comparisons of Tri+Qua+CB-HEM with using different settings (M=40, p=0.2 vs. M=5, p=0.8).

A (denoted by 'w/ A', finding the hard positive samples based on many small distance matrixes rather than a large distance matrix) reduce the training time by about 20% (from 120 minutes to 96 minutes for training an epoch) without degrading performance. When further adding the modification B (denoted by 'w/ AB', simplifying distance calculation), the performance keeps unchanged. For time consumption, it can only be reduced a little bit with the modification B, which may due to that the amount of distance calculation is relatively small and it is calculated on GPU.

2) Experiments With Contrastive Loss: To further verify the effectiveness of the proposed CB-HEM and PLB, we also conduct the experiments with Contrastive Loss (denoted as 'Cont'). The margin of contrastive loss is set to 1.2, and the settings of CB-HEM and PLB are the same as above ($M = 40$, $p = 0.2$, $PseudoN = 5$). The experimental results are shown in Fig. 10 (a). When training with Contrastive loss, our CB-HEM and PLB also can improve the performance by a lot. For example, CB-HEM and PLB improve the accuracy about 6% and 5% at FAR=1e-5, respectively. The improvements show the generality of our proposed CB-HEM and PLB.

3) How About Using a Small M and a Large r?: The parameters M and r are highly coupled. To select promising values for those two parameters in an efficient way, we first take experiments to determine the value of M and then search the best value for r. According to our experiments in Section IV-C, a large value for $M = 40$ and a small value for $r = 0.2$ are finally determined. How about the performance when using a small M and a large r? Here, we take a series of

TABLE VII

DEEP ANALYSIS ON PESUDO LARGE BATCH. ALL NETWORKS ARE TRAINED ON PRIVATE-IVS-TRAIN-S AND EVALUATED WITH PRIVATE-IVS-TEST

Method	Verification Rate (VR)		
	FAR=1e-6	FAR=1e-5	FAR=1e-4
Tri	82.00	90.40	95.38
Tri+PLB-Plain	80.76	90.26	95.68
Tri+PLB	84.16	91.76	96.50
Qua	80.60	90.88	95.90
Qua+PLB-Plain	80.96	90.98	96.24
Qua+PLB	83.66	92.24	96.78
Tri+Qua	84.30	92.54	96.64
Tri+Qua+PLB-Plain	83.66	92.24	96.78
Tri+Qua+PLB	85.60	92.66	97.02
Tri+Qua+CB-HEM	85.90	93.74	97.18
Tri+Qua+CB-HEM+PLB-Plain	88.40	94.90	97.66
Tri+Qua+CB-HEM+PLB	89.18	95.40	98.06

experiments (gradually change M from 40 to 5 while r from 0.2 to 1). The experimental results are shown in Fig. 10 (b). Generally, training the network with a larger M and a small r (e.g., $M = 40$ and $r = 0.2$) can achieve better performance than that of using a small M and a large r (e.g., $M = 5$ and $r = 1$). Intuitively, using a large M enlarges the selecting space of cross-batch hard triplets and using a small r can help the network to select the most difficult positive sample pairs for training. This may be the reason why using a large M and a small r can achieve better performance.

4) Deep Analysis on PLB: In this Section, a comprehensive study on PLB is conducted. We first analyze how much improvement PLB will bring to each method, with conducting the experiments with or without PLB settings (e.g., Tri vs. Tri+PLB, Qua vs. Qua+PLB and so on). Moreover, to further verify the effectiveness of PLB, a plain setting without $L^{bh}_{cross\_iter}$ is employed for comparison. We denote this plain setting as PLB-Plain. Actually, PLB-plain is simply aggregating the gradients from multiple iterations to update the network and there are no connections among different iterations. The experimental results are shown in Table VII. For our PLB, it can stably improve the performance on all methods (including Tri, Qua, Tri+Qua, Tri+Qua+CB-HEM). However, the performance is hardly improved by using PLB-Plain except for Tri+Qua+CB-HEM+PLB-Plain. For Tri+Qua+CB-HEM+PLB-Plain, using PLB-Plain can improve its performance because it can enlarge the selecting space for CB-HEM to select harder sample pairs for training. In this way, the performance improvements come form CB-HEM rather than PLB-Plain itself. This also shows simply that aggregating the gradients from multiple iterations/runs hardly improves the performance.

5) Visual Assessment: Fig. 11 and Fig. 12 shows the falsely classified images on Public-IvS and Private-IvS-Test datasets, respectively. In Private-IvS-Test dataset, the failures of false accept and false reject pairs may come from the low resolution, poor illumination, extreme pose, eyeglasses and so on. In Public-IvS, the failures come from two parts. One is external factors as explained above and the other is noises, where the labels of some pairs are wrong. For example, the first and second columns in Fig. 11 show the faces of the same identity, but the dataset annotates them as different identities.

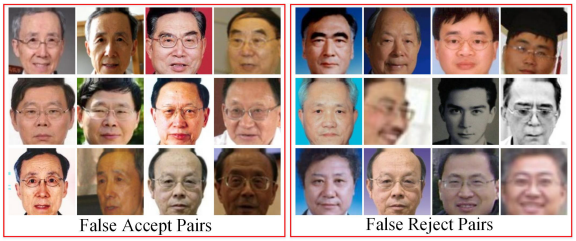


Fig. 11. Falsely classified images (both false accept and reject pairs) in Public-IvS-Test datasets at FAR=1e-5. Some noises of Public-IvS dataset are also shown in the figure.

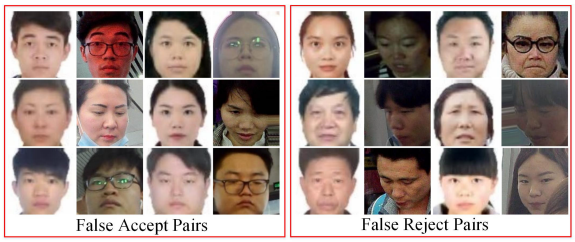


Fig. 12. Falsely classified images (both false accept and reject pairs) in Private-IvS datasets at FAR=1e-5.

V. CONCLUSION

ID vs. Spot face recognition plays an important role in our daily life. In this paper, CB-HEM and PLB have been proposed to train the network with very hard sample pairs, which improves the network's discriminative capability. Different from the previous B-HEM, the proposed CB-HEM can select hard sample pairs from past batches rather than only the current batch. For PLB, it virtually increases the batch size by updating the network once every few iterations with the accumulated gradients. With CB-HEM and PLB, the hard sample pairs can be selected from a large image space hundreds of times larger than the current batch space, which breaks through the limitation of GPU memory. Extensive experiments on IvS face datasets have verified the effectiveness of the proposed CB-HEM and PLB.

REFERENCES

[1] A. K. Jain and S. Z. Li, *Handbook of Face Recognition*. New York, NY, USA: Springer, 2011.
[2] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, 2015, pp. 1–12.
[3] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.
[4] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, vol. 2, no. 3, p. 7.
[5] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 212–220.
[6] R. Weng, J. Lu, and Y.-P. Tan, "Robust point set matching for partial face recognition," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1163–1176, Mar. 2016.
[7] J. Lu, G. Wang, and J. Zhou, "Simultaneous feature and dictionary learning for image set based face recognition," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 4042–4054, Aug. 2017.
[8] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.
[9] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[11] S. Z. Li, R. Chu, S. Liao, and L. Zhang, "Illumination invariant face recognition using near-infrared images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 627–639, Apr. 2007.

[12] 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, Jun. 2019.

[13] S. Yi, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. NIPS*, vol. 27, 2014, pp. 1–9.

[14] D. White, R. I. Kemp, R. Jenkins, M. Matheson, and A. M. Burton, "Passport officers' errors in face matching," *PLoS ONE*, vol. 9, no. 8, Aug. 2014, Art. no. e103510.

[15] Z. Dong, C. Jing, M. Pei, and Y. Jia, "Deep CNN based binary hash video representations for face retrieval," *Pattern Recognit.*, vol. 81, pp. 357–369, Sep. 2018.

[16] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Comput. Vis. Image Understand.*, vol. 189, Dec. 2019, Art. no. 102805.

[17] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," 2016, *arXiv:1607.08221*.

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

[19] 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. (CVPR)*, Jul. 2017, pp. 403–412.

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

[21] 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. (CVPR)*, Jun. 2016, pp. 761–769.

[22] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "MagFace: A universal representation for face recognition and quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14225–14234.

[23] 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. (CVPR)*, Jun. 2021, pp. 10492–10502.

[24] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[25] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[27] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[28] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Proc. ICB*, 2007, pp. 828–837.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1–9.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[31] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

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

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

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

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

[36] Z. Tan, Y. Yang, J. Wan, H. Hang, G. Guo, and S. Z. Li, "Attention-based pedestrian attribute analysis," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6126–6140, Dec. 2019.

[37] Z. Tan, J. Wan, Z. Lei, R. Zhi, G. Guo, and S. Z. Li, "Efficient group-N encoding and decoding for facial age estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2610–2623, Nov. 2018.

[38] G. Chen, J. Lu, M. Yang, and J. Zhou, "Learning recurrent 3D attention for video-based person re-identification," *IEEE Trans. Image Process.*, vol. 29, pp. 6963–6976, 2020.

[39] M. Tan and Q. V. Le, "EfficientNetV2: Smaller models and faster training," 2021, *arXiv:2104.00298*.

[40] 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.*, Jun. 2014, pp. 1701–1708.

[41] 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.

[42] B. Liu *et al.*, "Fair loss: Margin-aware reinforcement learning for deep face recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10051–10060.

[43] Y. Huang *et al.*, "CurricularFace: Adaptive curriculum learning loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5900–5909.

[44] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 542–5419.

[45] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2006–2014.

[46] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "AdaptiveFace: Adaptive margin and sampling for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11939–11948.

[47] 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.

[48] 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. (CVPR)*, Jun. 2020, pp. 8323–8332.

[49] 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. (ICCV)*, Oct. 2019, pp. 2326–2335.

[50] 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, 2020.

[51] V. Starovoitov, D. Samal, and B. Sankur, "Matching of faces in camera images and document photographs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Jun. 2000, pp. 2349–2352.

[52] V. Starovoitov and D. Samal, "Three approaches for face recognition," in *Proc. Int. Conf. Pattern Recognit. Image Anal.*, 2002, pp. 707–711.

[53] Y. Shi and A. K. Jain, "DocFace: Matching ID document photos to selfies," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–8.

[54] 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.

[55] V. Albiero *et al.*, "Identity document to selfie face matching across adolescence," 2019, *arXiv:1912.10021*.

[56] 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)*, Jun. 2006, pp. 1735–1742.

[57] 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. (CVPR)*, Jun. 2016, pp. 4004–4012.

[58] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Proc. NIPS*, 2016, pp. 1–9.

[59] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2612–2620.

[60] 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. (CVPR)*, Jun. 2019, pp. 5017–5025.

[61] 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. (CVPR)*, Jun. 2020, pp. 6388–6397.

[62] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. NIPS*, 2016, pp. 1–9.

[63] 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. (CVPR)*, Jun. 2019, pp. 5202–5211.

[64] 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. (ICCV)*, Oct. 2017, pp. 360–368.

[65] 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. (CVPR)*, Jun. 2020, pp. 3235–3244.

[66] Q. Qian *et al.*, "SoftTriple loss: Deep metric learning without triplet sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6450–6458.
[67] H. Shi *et al.*, "Embedding deep metric for person re-identification: A study against large variations," in *Proc. ECCV*, 2016, pp. 732–748.
[68] 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. (CVPR)*, Jun. 2016, pp. 761–769.
[69] A. Nech and I. Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3406–3415.
[70] 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," in *Proc. Workshop Faces 'Real-Life' Images, Detection, Alignment, Recognit.*, E. Learned-Miller, A. Ferencz, and F. Jurie, Eds. Marseille, France, Oct. 2008. [Online]. Available: https://hal.inria.fr/inria-00321923
[71] 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*, Sep. 2014, pp. 1–8.
[72] 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.
[73] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 1–9.
[74] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, 2019, pp. 1–12.
[75] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. NIPS*, 2008, pp. 1–8.

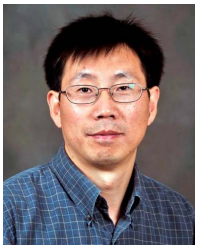
Zichang Tan received the B.E. degree from the Department of Automation, Huazhong University of Science and Technology (HUST), in 2016, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2021. Since July 2021, he has with Baidu Research as a Researcher. His main research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

Ajian Liu received the B.E. degree from the College of Physics and Information Engineering, Shanxi Normal University, Shanxi, China, in 2015, and the master's degree from the College of Information and Computer, Taiyuan University of Technology, Shanxi, in 2018. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Macau University of Science and Technology (MUST). 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 (NLPR), Institute of Automation, Chinese Academy of Science (CASIA), 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* (2020–2022), the Area Chair for ICME 2021 and 2020, the Senior Program Committee for AAAI 2021, and has served as a Co-Editor for special issues in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.

Hao Liu received the B.S. degree from the Beijing Institute of Technology (BIT) in 2016 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2021. He is currently a Senior Researcher with Tencent AI Lab. His research interests include computer vision and pattern recognition, especially with a focus on face recognition.

Zhen Lei (Senior Member, IEEE) received the B.S. degree in automation from the University of Science and Technology of China (USTC) in 2005 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010. He is currently a Professor with the Chinese Academy of Sciences. 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 the 2019 IAPR Young Biometrics Investigator Award. He has served as the Area Chair for the International Joint Conference on Biometrics in 2014; the IAPR/IEEE International Conference on Biometric in 2015, 2016, and 2018; and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015.



Guodong Guo (Senior Member, IEEE) received the B.E. degree in automation from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the University of Wisconsin, Madison, WI, USA.

He is currently the Head of the Institute of Deep Learning (IDL), 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 the Institute of Automation, Chinese Academy of Sciences; INRIA, Sophia Antipolis, France; Ritsumeikan University, Kyoto, Japan; and Microsoft Research, Beijing. He has 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 has published over 180 technical papers and he is the creator of the visual BMI (body mass index) estimator. His research interests include computer vision, biometrics, machine learning, and multimedia. He has received the North Carolina State Award for Excellence in Innovation in 2008, the New Researcher of the Year (2010–2011), and the Outstanding Researcher (2017–2018 and 2013–2014) at CEMR, WVU. He was selected the "People's Hero of the Week" by BSJB under Minority Media and Telecommunications Council (MMTC) in 2013. Two of his papers were selected as "The Best of FG'13" and "The Best of FG'15," respectively. He is an AE of several journals, including IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.

Stan Z. Li (Fellow, IEEE) received the B.Eng. degree from Hunan University, China, in 1982, the M.Eng. degree from the National University of Defense Technology, China, in 1985, and the Ph.D. degree from Surrey University, U.K., in 1991. He was a Researcher and the Director of the Center for Biometrics and Security Research, Institute of Automation, Chinese Academy of Sciences. He was a Researcher with Microsoft Research Asia and an Associate Professor with Nanyang Technological University, Singapore. He is currently a Chair Professor of artificial intelligence with Westlake University, China. He has published over 500 articles with Google Scholar index of over 47 000 and H-index of 127.