# Virtual Fully-Connected Layer: Training a Large-Scale Face Recognition Dataset with Limited Computational Resources

Pengyu Li[1], BiaoWang[1], Lei Zhang[1,2]
[1] Artificial Intelligence Center, DAMO Academy, Alibaba Group
[2]Department of Computing, Hong Kong Polytechnic University
lipengyu007@gmail.com, wangbiao225@foxmail.com, cslzhang@comp.polyu.edu.hk

## Abstract

*Recently, deep face recognition has achieved significant progress because of Convolutional Neural Networks (CNNs) and large-scale datasets. However, training CNNs on a large-scale face recognition dataset with limited computational resources is still a challenge. This is because the classification paradigm needs to train a fully-connected layer as the category classifier, and its parameters will be in the hundreds of millions if the training dataset contains millions of identities. This requires many computational resources, such as GPU memory. The metric learning paradigm is an economical computation method, but its performance is greatly inferior to that of the classification paradigm. To address this challenge, we propose a simple but effective CNN layer called the Virtual fully-connected (Virtual FC) layer to reduce the computational consumption of the classification paradigm. Without bells and whistles, the proposed Virtual FC reduces the parameters by more than 100 times with respect to the fully-connected layer and achieves competitive performance on mainstream face recognition evaluation datasets. Moreover, the performance of our Virtual FC layer on the evaluation datasets is superior to that of the metric learning paradigm by a significant margin. Our code will be released in hopes of disseminating our idea to other domains*[1].

## 1. Introduction

Recently, deep face recognition with Convolutional Neural Networks (CNNs) has achieved remarkable progress because of the explosion of large-scale training datasets. Guo et al. [3] released a dataset with almost 100 thousand identities in the academic field. In the industrial field, there are millions of identities used to train face recognition models. For instance, the face dataset produced by Google in 2015 had 200 million images consisting of 8 million different identities [15]. A large-scale training dataset helps a model obtain excellent performance, but it also challenges face recognition training paradigms.

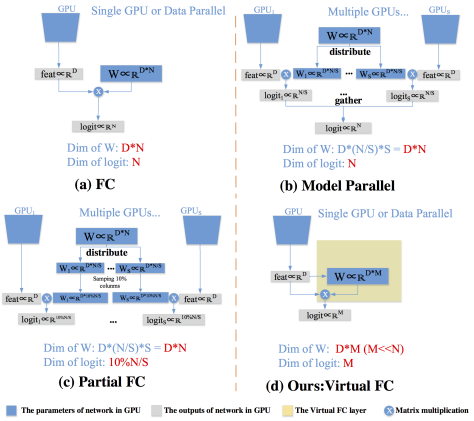


Figure 1. Comparison between FC (a), Model Parallel (b), Partial FC (c), and our Virtual FC (d). $D$ is the dimension of features, $N$ is the number of identities (categories), $S$ is the number of GPUs, and $M$ is a hyperparameter that can be set freely based on the balance between performance and computational resources. $M = 1\% \times N$ in this paper.

There are two elemental deep face recognition learning paradigms based on Convolutional Neural Networks [19]. One is learning with a classification loss function (e.g., the softmax loss function or ArcFace loss function) to optimize the similarity between samples and weight vectors [20, 1, 11, 22]. This classification paradigm has achieved state-of-the-art performance in face recognition fields. However, it needs to train a fully-connected (FC) layer as the category classifier, which leads to the following drawbacks: The FC layer is not necessary in inference, but it requires many computational resources in the training phase. Figure 2 shows that training on millions of identities requires the classification FC layer to include hundreds of

[1]https://github.com/pengyuLPY/Virtual-Fully-Connected-Layer.git

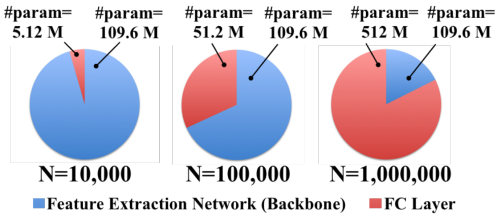


Figure 2. The parameters of the face recognition network in the classification paradigm. The FC layer requires considerable computational consumption, which may be even greater than the requirement of the backbone. The backbone is ResNet-101, and the feature dimension is 512 ($D = 512$) in this figure.

millions of parameters. Its parameters are much greater than those of the feature extraction network. The dimensions of its output are also in the millions. The cost of the storage and calculation of the FC layer easily exceeds current GPU capabilities (leading to an out of memory error, OOM) and results in training failure. The other paradigm is to leverage a metric learning loss function (e.g., the N-pair loss function or multi-similarity loss function) to optimize the similarity between samples [16, 18, 23, 19]. This paradigm addresses the drawbacks of the classification paradigm, but its performance is greatly inferior to that of the classification paradigm [19, 16].

Some technologies aim to solve the OOM problem in the classification paradigm with multiple GPUs, such as Model Parallel [7] and Partial FC [28]. Both of these split the FC layer into several parts, and each part is distributed to a respective GPU, as shown in Figure 1(b) and (c). Their solutions can train the dataset with millions of identities if there are enough GPUs. Figure 1 shows that the parameters and GPU memory are distributed but not reduced in the solutions. Thus, the solutions require many GPUs, and it is impossible to work with limited computational resources (e.g., a single GPU). The challenge of training large-scale face recognition datasets with limited training resources is still far from being solved.

To address these problems, we propose a simple but effective CNN layer called the **Virtual fully-connected (Virtual FC) layer** in this paper. The training pipeline of the Virtual FC layer is illustrated in Figure 3. The pipeline splits $N$ training identities into $M$ groups randomly. The identities from group $l$ share the $l$-th column in the projection matrix ($W$). The $l$-th column is called $anchor_l$. Because one group's identities share the anchors, the number of $W$ columns is reduced to $M$ from $N$ ($M << N$). The number of parameters in our Virtual FC is detailed in Figure 1 (d), and it is much less than that of other methods. Furthermore, the number of anchors ($M$) in our Virtual FC is a hyperparameter that is not limited by the batch size or number of identities in the mini-batch. It can be set freely based on the balance between performance and computational resources.

To optimize $W$, whose anchors are shared by the groups, we propose two novel types of anchors to constitute $W$. One is the **corresponding anchor**, and the other is the **free anchor**. If the mini-batch contains the identities from group $l$, $anchor_l$ belongs to the corresponding anchor and is estimated by a weighted average function. Otherwise, it is a free anchor and is estimated by the Stochastic Gradient Descent (SGD). The anchor type is adaptive in every training iteration.

The anchor would encounter conflict if the same group's identities were sampled to the mini-batch simultaneously because they would need to share the same anchor in this iteration. The straightforward strategy that avoids sampling identities from the same group cannot work because it means that intra-group identities have no chance to be optimized discriminatively. To eliminate anchor conflict effectively, we propose a **re-grouping strategy** in this paper.

Through our proposals, our Virtual FC layer can reduce the number of parameters by more than 100 times with respect to the FC layer and achieve competitive performance in typical face recognition evaluation datasets such as LFW [5], CFP [17], IJB-A [8], IJB-B [25], IJB-C [13], and MegaFace [6].

The main contributions of this paper can be summarized as follows:

1) To the best of our knowledge, we are the first to propose a solution for truly and significantly reducing the parameters in the classification paradigm to train large-scale face recognition datasets with limited computational resources (e.g., a single GPU).

2) We propose the Virtual fully-connected (Virtual FC) layer to train large-scale datasets with limited computational resources. The Virtual FC layer consists of corresponding anchors, free anchors, and a re-grouping strategy. The two types of anchors make it possible to optimize a $W$ whose columns are shared by groups. The re-grouping strategy is used to eliminate anchor conflict. Furthermore, the proposed Virtual FC layer is compatible with acceleration by Data Parallel [7] with multiple GPUs.

3) Without bells and whistles, the proposed Virtual FC reduces the parameters by more than 100 times to the fully-connected layer and achieves competitive performance. Moreover, the performance of our Virtual FC is superior to that of the metric learning paradigm by a significant margin.

## 2. Related Work

Face recognition is one of the most broadly researched topics in computer vision fields. Sun et al.[20] proposed Convolutional Neural Networks for solving the face recognition problem. Based on Convolutional Neural Networks, there are two elemental deep face recognition learning paradigms [19]. One is learning with a classification loss

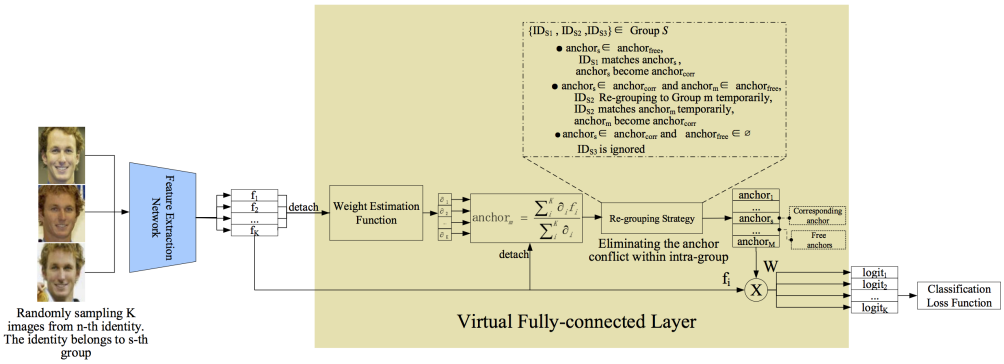


Figure 3. Training Pipeline of the Virtual FC Layer.

function to optimize the similarity between samples and weight vectors [20, 1, 11, 22]. The other is leveraging a metric learning loss function to optimize the similarity between samples [16, 18, 23, 19]. Because the number of pairs in the metric learning paradigm is limited by the mini-batch and the pair extraction strategy is tricky, its performance is inferior to that of the classification paradigm. However, the category classifier in the classification paradigm requires many computational resources in the training phase, which easily exceeds the current GPU capabilities and results in training failure. There is no perfect solution to balance the performance and the training resources.

Some technologies have tried to solve the OOM problem. Model Parallel [7] splits the final FC layer into several parts, and each part is distributed to a respective GPU. The logits predicted by all FC parts are synced to obtain a complete logit prediction. Its architecture is illustrated in Figure 1 (b). An et al. [28] proposed Partial FC based on Model Parallel to equally store the nonoverlapping linear transformation matrix on all GPUs in order. Each GPU is then accountable for calculating the sum of the dot product of the submatrix stored on its own and input features. After that, each GPU gathers the local sum from other GPUs to approximate the full-class softmax function. It reduces the dimension of the logits further by sampling columns of $W$ in each GPU. Its architecture is illustrated in Figure 1 (c). In summary, Model Parallel addresses the GPU memory limitation by distributing the FC layer to multiple GPUs, and Partial FC accelerates the process by reducing the sync logits. However, the parameters and the computational costs of their solutions are distributed but not reduced, as shown in Figure 1. Thus, their solutions require many GPUs, and it is impossible to work with limited computational resources (e.g., a single GPU).

Wen et al. [24] observed that each column in the projection matrix of the final FC layer indicates the centroid of a category representation. Liu et al. [12] proposed the transductive centroid projection (TCP) layer and used the centroid as the weight of unlabeled clustering data in every mini-batch. The number of columns in the TCP projection matrix is greatly limited by the batch size, which restricts its performance. However, the number of anchors in our Virtual FC is not limited by the batch size or number of identities in the mini-batch. It can be set freely based on the balance between performance and computational resources. This breakthrough helps the Virtual FC layer outperform the TCP by a significant margin.

## 3. Proposal: Virtual Fully-Connected Layer

Our Virtual FC layer is an improved fully-connected (FC) layer for training large-scale datasets with limited computational resources. The kernel of Virtual FC is the projection matrix $W \in \mathbb{R}^{D \times M}$. $D$ is the dimension of features. $M$ is the hyperparameter, which can be set freely based on the balance between performance and computational resources. With limited training resources (e.g., a single GPU), $M$ is set to be much less than the number of training identities ($N$). The output of the Virtual FC layer is calculated as for the FC layer:

$$y = W^T f + b \tag{1}$$

$y \in \mathbb{R}^M$ is the output, $b$ is the bias, and $f \in \mathbb{R}^D$ is the feature. $b$ is often set to zero in the representation learning field [1, 22, 11], so we ignore it in the following paper.

In the training pipeline of the Virtual FC layer, as Figure 3 shows, we split the training identities into $M$ groups randomly. The identities from group $l$ share the $l$-th column in $W$. This column is called $anchor_l$ in this paper. To optimize a $W$ whose anchors are shared by groups, we propose two novel types of anchors to constitute $W$ in the Virtual FC layer. One is the corresponding anchor, marked as $anchor_{corr}$. The other is the free anchor, marked as $anchor_{free}$. If the mini-batch contains identities from

group $l$, $anchor_l$ is of type $anchor_{corr}$. Otherwise, it is of type $anchor_{free}$. The anchor type is adaptive in every training iteration.

The anchor would encounter conflict if identities from the same group were sampled to the mini-batch simultaneously because they would need to share the same anchor in this iteration. To eliminate conflict, we propose a re-grouping training strategy in this paper. The re-grouping strategy achieves much better performance than the straightforward sampling strategy that avoids sampling identities from the same group to a mini-batch.

In the following sections 3.1 and 3.2, we introduce the proposed corresponding anchors and free anchors. Their optimization is formulated in section 3.3. To make our formulation brief and clear, we hypothesize that there is no anchor conflict in these three sections. The re-grouping strategy used to eliminate anchor conflict is introduced in section 3.4.

### 3.1. Corresponding Anchor

If the mini-batch contains identities from group $l$, $anchor_l$ belongs to the corresponding anchor in this iteration. Inspired by the observations of Wen et al. [24] and Liu et al. [12], namely, that each column in the projection matrix $W$ of the final fully-connected layer indicates the centroid of a category representation, we formulate our corresponding anchors as the following equation:

$$anchor_{corr,l} = \frac{\sum_{i=1}^{K} \alpha_{i,l} f_{i,l}}{\sum_{i}^{K} \alpha_{i,l}} \tag{2}$$

$f_{i,l}$ is the feature of the $i$-th image that belongs to group $l$. Because we hypothesize that there is no anchor conflict in this section, $\{f_{i,l}\}(i = 1, 2, ..., K)$ belong to a single identity. $\alpha_{i,l}$ is the attention estimation used to weight $f_{i,l}$. $\{\alpha_{i,l}\}$ can be estimated by the attention mechanism or set to be a constant value. If $\{\alpha_{i,l}\}(i = 1, 2, ..., k)$ is equal to a constant value, then $anchor_{corr,l}$ is the centroid of $\{f_{i,l}\}$. In the supplementary material, we show that the $\alpha_{i,l}$ estimated by a two-layer Multi-Layer Perceptron (MLP) [30] approximates a constant value, which means that $anchor_{corr}$ approximates a centroid. Based on this discovery, we set $\alpha_{i,l}$ to one and use the centroid as $anchor_{corr,l}$ directly in our experiments.

We illustrate the feature distribution and the centroids with a toy experiment in Figure 4. The training dataset is MNIST (N=10) [9], and the network is LeNet [2] (D=500) [10]. The left column is trained with a classical FC layer, the middle column is a Virtual FC (M=5), and the right column is a Virtual FC (M=2). The figure shows that the feature distribution of the Virtual FC layer is similar to that of the FC layer, and their accuracies are comparable.

[2]https://github.com/BVLC/caffe/blob/master/examples/mnist/

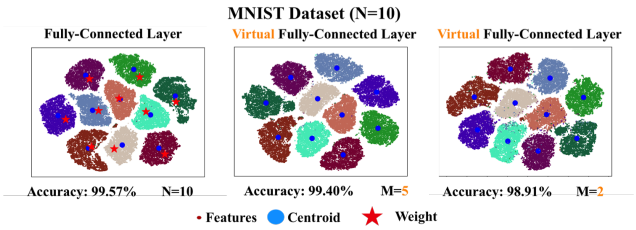


Figure 4. Toy Example: Training with LeNet on the MNIST dataset from scratch. The Virtual FC trains a feature extraction network. The feature extraction network is then fixed, and a linear classifier (N=10) is trained to classify the categories based on the features.

### 3.2. Free Anchor

The number of corresponding anchors is limited by the batch size. Because of the limitation of GPU memory, it is difficult to implement a large batch size. The limited number of $anchor_{corr}$ restricts the performance of our Virtual FC. To break this limitation and further improve the performance of Virtual FC, we propose novel free anchors in this section.

$anchor_l$ is a free anchor, marked as $anchor_{free,l}$, if there is no image from group $l$ in this iteration. $anchor_{free,l}$ cannot be calculated with Equation 2 because $f_{i,l} \in \emptyset$. In this paper, we optimize the free anchors with the SGD optimization method, which is detailed in section 3.3 below.

The free anchor is vital to our Virtual FC for the following reasons:

1) Free anchors break the limitation of the batch size as the classical FC layer does. This helps to disperse the inter-identity representations across the mini-batches.

2) Because free anchors are not limited by the batch size or number of identities in the mini-batch, the number of columns (M) can be set freely based on the balance between performance and computational resources.

3) The re-grouping strategy for eliminating anchor conflict would not work without free anchors. It would be a disaster, as section 3.4 discusses.

### 3.3. Optimization

Because the output of a Virtual FC layer belongs to $\mathbb{R}^M$, the ground truth of identities needs to be affiliated with the corresponding group IDs in the classification loss function. Because we hypothesize in this section that all the identities in the mini-batch are from different groups, the group ID $l$ is almost equivalent to the ground truth of identity.

The softmax loss function of our Virtual FC is shown in Equation 3. In addition, our Virtual FC is applicable to other classification loss functions, such as the ArcFace loss function [1], CosFace loss function [22], and SphereFace

loss function [11].

$$\mathbb{L} = -log(\frac{exp(y_l)}{\sum_i^M exp(y_i)}) \tag{3}$$

$y_i$ is the output of the Virtual FC layer, as shown in Equation 1. $l$ is the ground truth.

We investigate the gradients of our Virtual FC and its optimization based on Stochastic Gradient Descent (SGD). $f_{i,l}$ is detached from the neural network when it is used to estimate $anchor_{corr}$ in Equation 2. This detachment makes it easy to calculate the gradient and the error term. Based on the chain rule, the gradient of Virtual FC is shown in Equation 4, and its error term is shown in Equation 5.

$$\frac{\partial \mathbb{L}}{\partial W} = \frac{\partial \mathbb{L}}{\partial y}\frac{\partial y}{\partial W} \quad = \frac{\partial \mathbb{L}}{\partial y}\frac{\partial W^T f}{\partial W} = f(\frac{\partial \mathbb{L}}{\partial y})^T \tag{4}$$

$$\frac{\partial \mathbb{L}}{\partial f} = \frac{\partial \mathbb{L}}{\partial y}\frac{\partial y}{\partial f} = \frac{\partial \mathbb{L}}{\partial y}\frac{\partial W^T f}{\partial f} = W\frac{\partial \mathbb{L}}{\partial y} \tag{5}$$

Because $f$ is detached when it is used to estimate the anchor in Equation 2, the backward gradient and error term of the Virtual FC in Equations 4 and 5 are the same as those of the classical FC layer. This means that the Virtual FC can be embedded into the classification paradigm easily by merely substituting the last FC layer with the proposed Virtual FC layer.

Based on the SGD optimization method and the proposed corresponding anchors and the free anchors, the training iteration of the Virtual FC is formulated as the following set of equations:

**Forward**:

$$anchor^t = \begin{cases} Equation\ 2 & \text{anchor}^t \in anchor_{corr} \\ anchor^{t-1} & \text{anchor}^t \in anchor_{free} \end{cases} \tag{6}$$

$$anchor^0 = Scratch \tag{7}$$

$$W^t = [anchor_1^t, ..., anchor_M^t] \tag{8}$$

**Backward**:

$$W^{t+1} = W^t - \lambda^t \frac{\partial \mathbb{L}}{\partial W^t} \tag{9}$$

$$anchor_i^{t+1} = W_i^{t+1}, i = 1, 2, ..., M \tag{10}$$

$t$ denotes the iteration number, $i$ is the column number, and $\lambda^t$ is the learning rate used in SGD.

The optimization shows that $anchor_{corr,i}$ is estimated by Equation 2. $anchor_{free,j}$ is $W_j$, which is optimized by the SGD in the same way as the FC layer. Both $anchor_{corr}$ and $anchor_{free}$ deliver the error term to the feature extraction network (backbone) and make the learned representations compact in intra-identities and dispersed in inter-identities.

In addition, the type of anchor is determined by the sampling images in the mini-batch. This makes the anchor type adaptive in every training iteration.

### 3.4. Re-grouping Strategy

In sections 3.1, 3.2, and 3.3, we hypothesize that there is no anchor conflict in any mini-batch. A straightforward way to satisfy this hypothesis is to avoid sampling identities that are from the same group into a mini-batch. However, this leads to intra-group identities not being optimized to be dispersive. This shortcoming is a disaster for deep face recognition. To address this problem, we propose an effective re-grouping strategy in this section.

There are three states that an identity from the group $l$ could be in: 1) $anchor_l$ has not been matched yet. In this state, $anchor_l$ serves as the corresponding anchor for the identity. 2) $anchor_l$ has been matched by another identity from group $l$, but there are free anchors in the Virtual FC layer. In this state, we randomly select one free anchor, $anchor_{l'}$, and re-group the identity to the group $l'$. $anchor_{l'}$ will temporarily serve as the corresponding anchor for the identity in this iteration. After the updating of this iteration, both the identity's group and the anchor's parameters will be RECOVERED to their original ones. 3) $anchor_l$ has been used, and there is no free anchor left in the iteration. The identity and its images will be ignored. The re-grouping strategy based on these states is shown in Figure 3.

With the proposed re-grouping strategy, identities can be sampled randomly without considering their groups. Intra-group identities can be sampled simultaneously into a mini-batch, and their features can be optimized to be discriminative.

## 4. Experiment

The target of our Virtual FC layer is to train a large-scale face recognition dataset with limited computational resources. Therefore, we mainly compare our proposal with algorithms that could significantly reduce the parameters and computational costs, such as TCP and metric learning paradigms (i.e., the N-pair loss function [18] and the multi-similarity loss function [23]). We hypothesize that the performance of Model Parallel [7] and Partial FC [28] is close to the upper boundary obtained by the FC layer. However, their parameters and computational costs are distributed but NOT reduced. Thus, their solutions require many GPUs, and it is impossible to work with limited computational resources (e.g., a single GPU). Therefore, we do not take them into account for comparison.

In this section, we train the models on the largest-scale public dataset, MS-Celeb-1M [3]. The performance shown on the evaluation datasets, including CFP [17], LFW [5], IJB-A [8], IJB-B [25], IJB-C [13], and MegaFace [6],

proves the effectiveness of our Virtual FC. The theoretical analysis in Section 3 shows that our proposal can also work in an industrial dataset with millions of identities.

### 4.1. Dataset

In **MS-Celeb-1M**, there are almost 100 thousand global celebrities and 10 million images released. We clean the dataset by the automatic method proposed in [27]. The cleaned MS-Celeb-1M dataset in this paper contains 74,974 identities and 4.8 million images. **CASIA-WebFace** consists of 494,414 near-frontal faces of 10,575 subjects from the internet.

**CFP** consists of 10 folders, and each folder contains 350 same-person pairs and 350 different-person pairs for both frontal-frontal (CFP-FF) and frontal-profile (CFP-FP) experiments. **LFW** consists of 13,323 web photos of 5,749 celebrities, which are divided into 6,000 face pairs in 10 splits. In this paper, we follow the standard protocols of LFW and CFP and report their mean accuracy and the standard error of the mean.

The **IJB-A** dataset contains 5,397 images and 20,412 video frames split from 2,042 videos of 500 individuals. **IJB-B** and **IJB-C** are extensions of IJB-A. IJB-B contains 1,845 subjects with 21.8 K still images and 55 K frames from 7,011 videos. IJB-C contains 140,740 face images of 3,531 subjects. We evaluate the performance with their standard protocols. Because of the paper length limitation, we report the true acceptance rates (TARs) under a $10^{-2}$ false acceptance rate (FAR) in the paper. The top-K accuracy and the TARs under varying FARs are reported in the supplementary material.

The **MegaFace** dataset includes the probe and gallery set. The probe set is the FaceScrub dataset [14], which contains 100,000 images of 530 identities, and the gallery set consists of approximately 1,027,060 images from 690,572 different subjects. We report its rank1@$10^6$ accuracy[3], which is tested on the cleaned dataset[4].

The performance on **CALFW** [32], **CPLFW** [31], **SLLFW** [2], and **YTF** [26] is given in the supplementary materials because of the paper length limitation.

### 4.2. Implementation Details

Three backbones are trained in this paper to prove that Virtual FC can be embedded into different face recognition networks. They are CASIA-Net [29], ResNet-50 and ResNet-101 [4]. Their feature length is 512. We use the state-of-the-art loss function ArcFace [1] in this paper. Other classification loss functions, such as CosFace [22], NormFace [21], SphereFace [11], and CircleLoss [19], can also work with the Virtual FC. However, the loss function is not in the scope of this paper.

[3]http://megaface.cs.washington.edu/dataset/download/content/devkit.zip
[4]https://github.com/deepinsight/insightface

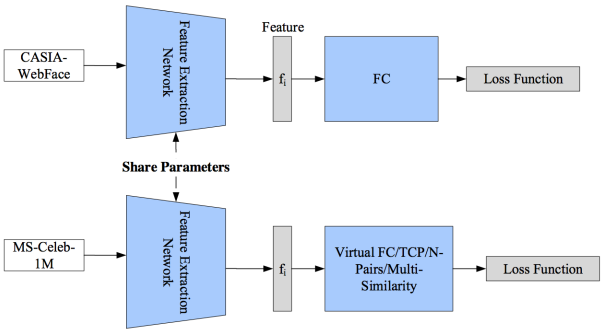


Figure 5. Multi-task training strategy

For CASIA-Net, we randomly sample 96×96 regions from the aligned 100×100 face images for data augmentation. The image intensities are linearly scaled to the range [−1, 1]. The network is trained for 30 epochs. The learning rate is 0.1 and decays 10 times at the $20_{th}$, $27_{th}$ and $29_{th}$ epochs. The momentum is 0.9, and the weight decay is 0.0005.

For ResNet-50 and ResNet-101, we resize the face images to 224×224. The image intensities are scaled to [-1, 1]. The networks are optimized with SGD for 16 epochs. The learning rate is 0.01 and decays ten times at the $12_{th}$, $14_{th}$ and $15_{th}$ epochs. The momentum is 0.9, and the weight decay is 0.0005.

The number of identities of MS-Celeb-1M is 74,974 (i.e., $N = 74,974$). We sample 7 images per identity into a mini-batch (i.e., $K = 7$). The number of groups and the number of anchors are 1000 (i.e., $M = 1000$), which are approximately $1\% \times N$. The number of $anchor_{corr}$s is 100, which is approximately $0.1\% \times N$. The influence of $K$ and $M$ will be discussed in the ablation study.

TCP requires a fully-connected layer in its pipeline. It is challenging for the metric learning loss functions (i.e., N-pair, multi-similarity) to converge without the classification loss function. All of them require an FC layer followed by a classification loss function in their training pipelines. For a fair comparison, we use the multi-task training strategy in the experiments. The pipeline of multi-task is shown in Figure 5. The lower boundary is trained on the CASIA-WebFace dataset ($N \approx 10,000$) with the FC layer. The upper boundary is trained on the CASIA-WebFace dataset and MS-Celeb-1M dataset ($N \approx 100,000$) with the FC layer. The experiment of training on a single task with our Virtual FC is detailed in the ablation study.

### 4.3. Comparison with Other Methods

We compare our Virtual FC with other candidate solutions in Table 1. The backbone is CASIA-Net, ResNet-50 or ResNet-101. The table shows the following:

| Networks/Methods | | Evaluation Dataset | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | LFW | CFP-FF | CFP-FP | IJB-A@10-2 | IJB-B@10-2 | IJB-C@10-2 | MegaFace |
| CASIA-Net | Lower boundary | 98.05±0.64 | 98.56±0.54 | 91.66±1.91 | 89.90 | 88.40 | 89.94 | 73.14 |
| | Upper boundary | 99.07±0.44 | 99.40±0.26 | 93.87±1.32 | 93.89 | 93.54 | 94.37 | 80.52 |
| | N-pair [18] | 98.68±0.37 | 99.03±0.30 | 92.70±1.51 | 85.65 | 87.25 | 88.69 | 81.11 |
| | Multi-similarity [23] | 98.61±2.47 | 99.03±0.35 | 92.11±1.23 | 85.53 | 85.21 | 85.90 | 78.67 |
| | TCP [12] | 98.73±0.53 | 99.03±0.47 | 92.71±1.74 | 89.75 | 90.32 | 91.86 | 80.08 |
| | ours: Virtual FC | **98.75±0.27** | **99.07±0.30** | **93.04±1.84** | **91.77** | **90.78** | **92.21** | **81.44** |
| ResNet-50 | Lower boundary | 97.88±0.61 | 99.11±0.39 | 93.47±1.41 | 90.59 | 91.26 | 92.78 | 79.28 |
| | Upper boundary | 99.55±0.24 | 99.91±0.37 | 96.97±0.91 | 97.33 | 96.71 | 97.19 | 97.63 |
| | N-pair [18] | 98.33±0.60 | 98.80±0.54 | 92.74±1.97 | 85.32 | 88.46 | 90.08 | 82.56 |
| | Multi-similarity [23] | 98.33±0.60 | 98.74±0.50 | 93.17±1.73 | 83.49 | 88.30 | 90.06 | 76.88 |
| | TCP [12] | 98.95±0.27 | 99.44±0.36 | 94.30±1.08 | 85.53 | 90.35 | 92.08 | 88.18 |
| | ours: Virtual FC | **99.32±0.27** | **99.73±0.33** | **95.77±1.11** | **92.83** | **93.21** | **94.53** | **93.18** |
| ResNet-101 | Lower boundary | 98.29±0.57 | 99.07±0.34 | 93.57±1.32 | 90.33 | 91.19 | 92.38 | 76.83 |
| | Upper boundary | 99.53±0.27 | 99.91±0.29 | 97.45±0.86 | 97.81 | 97.01 | 97.61 | 98.39 |
| | N-pair [18] | 98.33±0.60 | 98.79±0.34 | 92.86±1.25 | 83.55 | 86.57 | 88.72 | 75.33 |
| | Multi-similarity [23] | 97.82±0.29 | 98.86±0.43 | 92.76±1.56 | 82.02 | 85.56 | 87.50 | 75.70 |
| | TCP [12] | 99.30±0.28 | 99.63±0.27 | 95.77±1.09 | 89.23 | 92.67 | 94.22 | 92.41 |
| | ours: Virtual FC | **99.38±0.38** | **99.61±0.31** | **95.55±1.42** | **93.69** | **94.05** | **95.30** | **94.04** |

Table 1. Comparison with other methods

1) Our Virtual FC surpasses the lower boundary and all other candidate solutions consistently and significantly. It also achieves comparable performance to that of the upper boundary with 1% computational resources of the FC layer.

2) The superiority of our Virtual FC is more significant in complex neural network structures (e.g., ResNet50 and ResNet101) than in simple structures (CASIA-Net). For instance, the Virtual FC layer improves the performance of IJB-C from 92.08% (TCP) to 94.53% in ResNet50 and from 94.22% to 95.30% in ResNet101. The improvement is much greater than that in CASIA-Net (91.86% to 92.21%). So are the performance improvements on the other evaluation datasets.

### 4.4. Ablation Study

**The influence of the sampling image number ($K$) per identity in a mini-batch.** We study the influence of $K$ with CASIA-Net. All the implementation details are the same as those introduced in section 4.2 except for $K$. The performance is shown in Table 2. The table shows that the performance for all $K$ is higher than the lower boundary, and the performance is almost equivalent when $K \geq 5$. The performance of Virtual FC is insensitive to $K$ if $K \geq 5$.

| $K$ | Evaluation Dataset | | | | | | |
|---|---|---|---|---|---|---|---|
| | LFW | CFP-FF | CFP-FP | IJB-A | IJB-B | IJB-C | MegaFace |
| Lower boundary | 98.05±0.64 | 98.56±0.54 | 91.66±1.91 | 89.90 | 88.40 | 89.94 | 73.14 |
| Upper boundary | 99.07±0.44 | 99.40±0.26 | 93.87±1.32 | 93.89 | 93.54 | 94.37 | 80.52 |
| K=2 | 98.60±0.48 | 99.00±1.62 | 92.20±1.57 | 90.24 | 90.22 | 91.57 | 80.30 |
| K=5 | 98.60±0.56 | **99.19±0.45** | 93.20±1.24 | 91.73 | 91.20 | 92.50 | 80.31 |
| K=7 | **98.75±0.27** | 99.07±0.30 | 93.04±1.84 | 91.77 | 90.78 | 92.21 | **81.44** |
| K=10 | 98.72±0.52 | 99.00±0.27 | **93.34±1.39** | **92.11** | **91.88** | **93.13** | 80.75 |

Table 2. The influence of $K$. The performance of Virtual FC is insensitive to $K$ if $K \geq 5$.

**The influence of corresponding anchors and free anchors.** By changing the number of anchors ($M$), we study the influence of corresponding anchors and free anchors with CASIA-Net in Table 3. The table shows that the performance of our Virtual FC is improved with increasing $M$. There are two types of anchors that lead to an increase in $M$. 1) Corresponding anchors. If $M \leq 0.1\% \times N$, then there are only corresponding anchors in the Virtual FC. In these cases, our Virtual FC degrades to the TCP. The performance improvement that is caused by $M$ increasing to 0.1%×N from 0.01%×N proves that more corresponding anchors help Virtual FC/TCP achieve a better performance. However, the number of corresponding anchors is limited by the batch size. 0.1%×N is the most we can employ in our learning platform. This limitation restricts the performance of TCP. 2) Free anchors. If $M \geq 1\% \times N$, then there are free anchors in Virtual FC. The number of corresponding anchors is fixed to 0.1%×N because of the batch size limitation. The free anchors break this limitation and increase M to 1%×N or more. The improvement when $M \geq 1\% \times N$ proves the importance of the proposed free anchors.

**The influence of the re-grouping strategy.** There may be anchor conflicts in our Virtual FC layer, as introduced in section 3. There are two methods to eliminate the conflicts. One is our proposed re-grouping strategy. The other is a straightforward sampling strategy that avoids sampling identities that belong to the same group into a mini-batch simultaneously. We analyzed why the straightforward sampling strategy does not work in Section 3.4. We perform the experiments in Table 4[5] to prove this. The table shows that the re-grouping strategy has a limited improvement for CASIA-Net, but the performance for ResNet-50 and ResNet-101 drops considerably without it. We think this

[5]MegaFace is shown in the supplementary materials because of the paper length limitation.

| $M$ | Evaluation Dataset | | | | | | |
|---|---|---|---|---|---|---|---|
| | LFW | CFP-FF | CFP-FP | IJB-A | IJB-B | IJB-C | MegaFace |
| Lower boundary | 98.05±0.64 | 98.56±0.54 | 91.66±1.91 | 89.90 | 88.40 | 89.94 | 73.14 |
| Upper boundary | 99.07±0.44 | 99.40±0.26 | 93.87±1.32 | 93.89 | 93.54 | 94.37 | 80.52 |
| TCP [12] (M=0.1%×N) | 98.73±0.53 | 99.03±0.47 | 92.71±1.74 | 89.75 | 90.32 | 91.86 | 80.08 |
| M=0.01%×N | 94.67±1.26 | 99.06±0.41 | 92.37±1.36 | 89.61 | 88.57 | 89.72 | 81.21 |
| M=0.1%×N | 98.46±0.55 | 99.11±0.51 | 92.50±1.80 | 91.42 | 90.69 | 92.15 | 81.26 |
| M=1%×N* | 98.75±0.27 | 99.07±0.30 | 93.04±1.84 | 91.77 | 90.78 | 92.21 | 81.44 |
| M=10%×N* | 98.65±0.61 | **99.24±0.42** | **93.27±1.23** | **92.31** | 92.01 | 93.26 | 82.29 |
| M=100%×N* | **98.75±0.49** | 99.20±0.41 | 92.99±1.39 | 91.12 | **92.17** | **93.42** | **83.07** |

Table 3. The influence of corresponding anchors and free anchors. The performance of our Virtual FC improves with increasing M. A * means the Virtual FC in this row contains both corresponding anchors and free anchors. Otherwise, there are only corresponding anchors in the rows.

is because ResNets have massive parameters and capacity. If the straightforward sampling strategy is used, their excellent capacity makes them easily overfit in discriminating a group but not identities. CASIA-Net does not have the capacity of ResNet, and it would not overfit. The re-grouping strategy is essential to our Virtual FC layer, especially for complex backbones.

| Network | Strategy | LFW | CFP-FF | CFP-FP | IJB-A | IJB-B | IJB-C |
|---|---|---|---|---|---|---|---|
| CASIA-Net | Sampling | 98.52±1.90 | 98.94±0.27 | 92.76±1.44 | **92.14** | 90.34 | 91.77 |
| | Re-grouping | **98.75±0.27** | **99.07±0.30** | **93.04±1.84** | 91.77 | **90.78** | **92.21** |
| ResNet-50 | Sampling | 98.33±0.60 | 99.22±0.41 | 94.31±1.51 | 89.90 | 90.20 | 92.05 |
| | Re-grouping | **99.32±0.27** | **99.73±0.33** | **95.77±1.11** | **92.83** | **93.21** | **94.53** |
| ResNet-101 | Sampling | 98.64±0.47 | 98.67±0.56 | 92.59±1.54 | 87.16 | 88.10 | 89.60 |
| | Re-grouping | **99.38±0.38** | **99.61±0.31** | **95.55±1.42** | **93.69** | **94.05** | **95.30** |

Table 4. Importance of the proposed re-grouping strategy. The re-grouping strategy is essential to our Virtual FC, especially for complex feature extraction networks.

**Training Virtual FC with single-task learning.** We train our Virtual FC as a multi-task learning strategy in the previous experiments because the methods we compare cannot be trained as a single task. In Table 5, we train our Virtual FC as a single task (MS-Celeb-1M) and compare it with multi-task learning. For a fair comparison with multi-task learning, the networks are pretrained by CASIA-WebFace with an FC layer (N=10,000) and fine-tuned by MS-Celeb-1M with a Virtual FC layer (M=1000).

| Network | Task | LFW | CFP-FF | CFP-FP | IJB-A | IJB-B | IJB-C |
|---|---|---|---|---|---|---|---|
| CASIA-Net | Lower boundary | 98.05±0.64 | 98.56±0.54 | 91.66±1.91 | 89.90 | 88.40 | 89.94 |
| | Multi-task | 98.75±0.27 | 99.07±0.30 | 93.04±1.84 | 91.77 | 90.78 | 92.21 |
| | Single-task | 98.64±0.46 | 98.79±0.42 | 91.70±1.68 | 90.32 | 90.40 | 91.88 |
| ResNet-50 | Lower boundary | 97.88±0.61 | 99.11±0.39 | 93.47±1.41 | 90.59 | 91.26 | 92.78 |
| | Multi-task | 99.32±0.27 | 99.73±0.33 | 95.77±1.11 | 92.83 | 93.21 | 94.53 |
| | Single-task | 98.23±0.98 | 99.54±0.46 | 93.99±1.38 | 93.58 | 93.36 | 94.47 |
| ResNet-101 | Lower boundary | 98.29±0.57 | 99.07±0.34 | 93.57±1.32 | 90.33 | 91.19 | 92.38 |
| | Multi-task | 99.38±0.38 | 99.61±0.31 | 95.55±1.42 | 93.69 | 94.05 | 95.30 |
| | Single-task | 98.33±0.60 | 99.56±0.40 | 94.36±1.33 | 93.68 | 93.87 | 95.08 |

Table 5. Fine-tuning a network (lower boundary) pretrained on a small-scale dataset (CASIA-WebFace) as a single task with Virtual FC.

Table 5 shows that 1) Single-task learning surpasses the pretrained model (lower boundary) by a significant margin. 2) Single-task learning with our Virtual FC surpasses TCP [12], N-pair [18], and multi-similarity [23] in Table 1. All of them are trained with the multi-task learning strategy, which has better performance than the single-task strategy. 3) The performance of multi-task learning surpasses that of single-task learning. Multi-task learning is an effective method to improve the performance of our Virtual FC layer.

We train all the methods in Table 1 as a single task from scratch. N-pair [18] and multi-similarity [23] fail to converge in our experiments. We modify the architecture of TCP to allow it to be trained as a single task. All the performances are shown in Table 6. The table shows that our Virtual FC surpasses the other methods by a large margin. However, the performance degrades a lot when it is trained from scratch. We think this is because the intra-identity features are dispersed when it is trained with single-task learning from scratch. This makes our corresponding anchor generation fail and limits the performance of the Virtual FC layer.

| Method | Backbone | LFW | CFP-FF | CFP-FP | IJB-A | IJB-B | IJB-C |
|---|---|---|---|---|---|---|---|
| N-Pair | CASIA-Net | No Convergence | | | | | |
| Multi-Similarity | | No Convergence | | | | | |
| TCP | | 93.68±1.02 | 93.13±1.19 | 83.61±1.58 | 54.38 | 68.33 | 80.81 |
| Virtual FC | | 94.27±0.60 | 96.44±0.51 | 88.36±1.38 | 77.22 | 82.94 | 85.01 |
| | ResNet-50 | 96.57±0.85 | 98.05±0.54 | 88.07±1.92 | 86.43 | 87.37 | 89.37 |
| | ResNet-101 | 98.15±1.50 | 98.58±0.37 | 90.07±1.75 | 87.87 | 88.19 | 90.81 |

Table 6. Training the network as a single task from scratch with Virtual FC.

Tables 5 and 6 show that our Virtual FC can work well as a multi-task learning strategy or single-task learning from a pretrained model. Its performance would have a significant drop if it were trained as a single task from scratch. It would be our future work to address this limitation. However, it is easy to obtain a model that has been pretrained on a small-scale dataset or to train the model with the multi-task learning strategy (one task is trained on a small-scale dataset with the FC layer, and the other task is trained on a large-scale dataset with our Virtual FC layer). Our work will inspire the academic field and industrial field to train large-scale face recognition datasets with limited training resources.

## 5. Conclusion

This paper proposes a simple but effective Virtual fully-connected (Virtual FC) layer to train large-scale face recognition datasets in a classification paradigm with limited computational resources. The proposed Virtual FC reduces the parameters by 100 times with respect to the fully-connected layer and achieves competitive performance. Moreover, the performance of our Virtual FC is superior to that of the metric learning paradigm by a significant margin.

## References


[1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 4321, 4323, 4325, 4326

[2] Weihong Deng, Jiani Hu, Nanhai Zhang, Binghui Chen, and Jun Guo. Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership. *Pattern Recognition*, 66:63–73, 2017. 4326

[3] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016. 4321, 4326

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 630–645. Springer, 2016. 4326

[5] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008. 4322, 4326

[6] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882, 2016. 4322, 4326

[7] Nikhil Ketkar. Introduction to pytorch. In *Deep learning with python*, pages 195–208. Springer, 2017. 4322, 4323, 4325

[8] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1931–1939, 2015. 4322, 4326

[9] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 4324

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4324

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 4321, 4323, 4325, 4326

[12] Yu Liu, Guanglu Song, Jing Shao, Xiao Jin, and Xiaogang Wang. Transductive centroid projection for semi-supervised large-scale recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–86, 2018. 4323, 4324, 4327, 4328

[13] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018. 4322, 4326

[14] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347. IEEE, 2014. 4326

[15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 4321

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 4322, 4323

[17] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 4322, 4326

[18] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016. 4322, 4323, 4325, 4327, 4328

[19] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 4321, 4322, 4323, 4326

[20] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015. 4321, 4322, 4323

[21] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 4326

[22] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 4321, 4323, 4325, 4326

[23] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 4322, 4323, 4325, 4327, 4328

[24] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 4323, 4324

[25] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017. 4322, 4326

[26] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011. 4326

[27] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018. 4326

[28] An Xiang, Zhu Xuhan, Xiao Yang, Wu Lan, Zhang Ming, Gao Yuan, Qin Bin, Zhang Debing, and year=2020 Ying, Fu journal=arXiv preprint arXiv:2010.05222. Partial fc: Training 10 million identities on a single machine. 4322, 4323, 4325

[29] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 4326

[30] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2019. 4324

[31] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5, 2018. 4326

[32] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017. 4326