Noise-Tolerant Paradigm for Training Face Recognition CNNs

Wei Hu[1] Yangyu Huang[2*] Fan Zhang[1] Ruirui Li[1]
[1]Beijing University of Chemical Technology, China [2]Yunshitu Corporation, China
[1]{huwei,zhangf,liruirui}@mail.buct.edu.cn
[2]yangyu.huang.1990@gmail.com

Abstract

Benefit from large-scale training datasets, deep Convolutional Neural Networks(CNNs) have achieved impressive results in face recognition(FR). However, tremendous scale of datasets inevitably lead to noisy data, which obviously reduce the performance of the trained CNN models. Kicking out wrong labels from large-scale FR datasets is still very expensive, although some cleaning approaches are proposed. According to the analysis of the whole process of training CNN models supervised by angular margin based loss(AM-Loss) functions, we find that the θ distribution of training samples implicitly reflects their probability of being clean. Thus, we propose a novel training paradigm that employs the idea of weighting samples based on the above probability. Without any prior knowledge of noise, we can train high performance CNN models with large-scale FR datasets. Experiments demonstrate the effectiveness of our training paradigm. The codes are available at https://github.com/huangyangyu/NoiseFace.

1. Introduction

Large-scale datasets are crucial for training deep CNNs in FR, and the scale of training datasets is growing tremendously. For example, a widely used FR training dataset, MS-Celeb-1M [11], contains about 100K celebrities and 10M images. However, a previous work [42] points out that a million scale FR dataset typically has a noise rate higher than 30% (about 50% in the original MS-Celeb-1M). The presence of noisy training data may adversely affect the final performance of trained CNNs. Though a recent work [35] reports that deep CNNs still perform well even on noisy datasets containing sufficient clean data, this conclusion cannot be transferred to FR, and experiments demonstrate that noisy data apparently decrease the performance of the trained FR CNNs [42].

Large-scale datasets with high-quality label annotations are very expensive to obtain. Cleaning large-scale FR datasets with automatic or semi-automatic approaches [11, 49, 5] cannot really solve this problem. As can be seen, existed large-scale FR datasets, such as MS-Celeb-1M and MegaFace [19], still consist considerable incorrect labels. To obtain a noise-controlled FR dataset, manual annotation is inevitable. Although an approach is introduced to effectively build a high-quality dataset IMDB-Face [42], it actually further demonstrates the difficulties of obtaining a large-scale well-annotated FR dataset. For example, it took 50 annotators one month to clean the IMDB-Face dataset, which only contains 59K celebrities and 1.7M images.

Numerous training approaches have been investigated aiming to train classification CNNs with noisy datasets [31, 50, 38, 10, 41, 12, 21, 9], but most of them are not suitable for training FR models, because of the special characters of FR datasets (discussed in the Section 2). Recently, weighting training samples is a promising direction [18, 13, 26] to deal with noisy data. However, extra datasets or complex networks are required in these approaches, limiting the use of them in FR.

The CNN models in FR are usally trained with loss functions, which aim to maximize inter-identity variation and minimize intra-identity variation under a certain metric space. Very recently, some angular margin based loss(AM-Loss for short in the paper) functions [24, 5, 45, 43] are proposed and achieve the state-of-the-art performance.

In this paper, we propose a noise-tolerant paradigm to learn face features on a large-scale noisy dataset directly, different from other related approaches [49, 5] which aim to clean the noisy dataset firstly. When training an AM-Loss supervised CNN model, the θ histogram distribution of training samples(the θ distribution for short, described in Section 3.2) is employed to measure the possibility that a sample is correctly labeled, and this possibility is then used to determine the training weight of the sample. Throughout the training process, the proposed paradigm can alleviate the impact of noisy data by dynamically adjusting the weight of samples, according to their θ distribution at that time.

To summarize, our major works are as follows:

1. We observe that the θ value of a clean sample has high-

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.01216

er probability to be smaller than that of a noisy sample for an AM-Loss function. In other word, the possibility that a sample is clean can be dynamically reflected by its position in the θ distribution.

2. Based on the above observation, we employ the idea of weighting training samples, and present a novel noise-tolerant paradigm to train FR CNN models with a noisy dataset end-to-end. Without any prior knowledge of noise in the training dataset (noise rate, small clean sets, etc.), the models can achieve comparable, or even better performance, compared with the models trained by traditional methods with the same dataset without noisy samples.

3. Although many approaches are proposed to train classification models with noisy datasets, none of them aims to train FR CNN models. To the best of our knowledge, the proposed paradigm is the first to study the method to significantly eliminate adverse effects of extremely noisy data with deep CNN models in FR. Our paradigm is also the first to estimate the noise rate of a noisy dataset accurately in FR. Furthermore, our trained models can also achieve good performance on clean datasets too.

2. Related Works

2.1. Training with Noisy Data

Learning with noisy datasets has been widely explored in image classification training [7]. In classic image classification datasets, the real-world noisy labels exhibit multimode characteristics. Therefore, many approaches use predefined knowledge to learn the mapping between noisy and clean annotations, and focus on estimating the noise transition matrix to remove or correct mis-labeled samples [27, 23, 30]. Recently, it has also been studied in the context of deep CNNs. [50] relies on manually labeling to estimate the matrix. [38, 10] add layer in CNN models to learn the noise transition matrix. [41, 14] use a small clean dataset to learn a mapping between noisy and clean annotations. [31, 9] use noise-tolerant loss functions to correct noisy labels. Li *et al*. [21] construct a knowledge graph to guide the learning process. Han *et al*. [12] propose a human-assisted approach which incorporates an structure prior to derive a structure-aware probabilistic model. Different from the common classification datasets, FR datasets always contain a very large number of classes(persons), but each class contains relatively small number of images, making it difficult to find relationship patterns from noisy data. Furthermore, noisy labels behave more like independent random outliers in FR datasets. Therefore, the transition matrix or the relationship between noisy and clean labels is very hard to be estimated from FR datasets.

Some approaches attempt to update the trained CNNs only with separated clean samples, instead of correcting the noisy labels. A Decoupling technique [26] trains two CNN models to select samples that have different predictions from these two models, but it cannot process heavy noisy datasets. Very recently, weighting training samples becomes a hot topic to learn with noisy datasets.

2.1.1 Weighting Training Samples

Weighting training samples is a well studied technique and can be applied to adjust the contributions of samples for training CNN models [8, 22, 20]. Huber loss [8] reduces the contribution of hard samples by down-weighting the loss of them. In contrast, Focal loss [22] adds a weighting factor to emphasize hard samples for training high accurate detector. Multiple step training is adopted in [20] to encourage learning easier samples first.

The idea of weighting training samples is employed to train models with noisy datasets too, since clean/noisy samples are usually corresponding to easy/hard samples. The key of weighting samples is an effective method to measure the possibility that a sample is easy/clean. Based on Curriculum learning [2], MentorNet [18] and Coteaching [13] try to select clean samples using the small-loss strategy, but the noise level should be provided in advanced in [13], and a small clean set and a pre-training model are suggested in [18]. Ren *et al*. [34] also employ the clean validation set to help learning sample weights. Although FR can be regarded as a classification problem, its small subset/validation dataset is usually uncertain to be clean, even not available at all. To conclude, weighting samples is a promising direction to train CNN models with noisy datasets, but estimating sample weights usually requires complex techniques and extra knowledge.

2.2. Loss functions in FR

Deep face recognition has been one of the most active field in these years. Usually, FR is trained as a multi-class classification problem in which the CNN models are usually supervised by the softmax loss [40, 39, 46]. Some metric learning loss functions, such as contrastive loss [51, 4], triplet loss [15, 36] and center loss [47], are also applied to boost FR performance greatly. Other loss functions [6, 53] also demonstrate effective performance on FR. Recently, some normalization [25, 44, 54] and angular margin [24, 5, 43, 45] based methods are proposed and achieve outperforming performance, and get more attention.

3. Preliminaries

3.1. Angular Margin based Losses

Recently, the angular margin based loss functions, which have intrinsic consistency with Softmax loss, greatly improves the performance of FR. The traditional Softmax loss is presented as

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i+\mathbf{b}_{y_i}}}{\sum_{j=1}^{C} e^{\mathbf{W}_j^T\mathbf{x}_i+\mathbf{b}_j}}, \quad (1)$$

where $\mathbf{x}_i$ denotes the feature of the i-th sample which belongs to the y_i-th class. $\mathbf{W}_j$ denotes the j-th column of the weights $\mathbf{W}$ in the layer and $\mathbf{b}$ is the bias term. N and C is the batch size and the class number. In all AM-Losses, the bias $\mathbf{b}_j$ is fixed to be 0 and $\|\mathbf{W}_j\|$ is set to 1, then the target logit [32] can be reformulated as

$$\mathbf{W}_j^T\mathbf{x}_i = \|\mathbf{x}_i\| cos\ \theta_{i,j}, \quad (2)$$

where $\theta_{i,j}$ is the angle between $\mathbf{W}_j$ and $\mathbf{x}_i$. We can further fix $\|\mathbf{x}_i\| = s$, and the Softmax loss can be reformulated as

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\ cos\ \theta_{i,y_i}}}{\sum_{j=1}^{C} e^{s\ cos\ \theta_{i,j}}}, \quad (3)$$

and we refer this loss as L2-Softmax in this paper. Actually, other AM-Losses have similar loss functions, but with slightly different decision boundaries.

In all AM-Losses, $\mathbf{W}_j$ can be regarded as the anchor of the j-th class. During training, the angle $\theta_{i,j}$ will be minimized for an input feature $\mathbf{x}_i$ belonging to the j-th class, and the angle $\theta_{i,k(k\neq j)}$ will be maximized at the same time. As discussed in [5], the $\theta_{i,j}$ of an input $\mathbf{x}_i$ belonging to the j-th class reflects the difficulty of training the corresponding sample for a full trained CNN model, and the θ distribution of all training samples can implicitly demonstrate the performance of the model. We firstly investigate the effect of noisy data on the θ distributions.

3.2. Effect of Noise on θ Distributions

To investigate the effect of noise, WebFace-Clean[1] containing 10K celebrities and 455K images is chosen as the clean dataset. It is cleaned by manually removing incorrect images from the original CASIA-WebFace [51] containing 494K images. [42] estimates that there are about 9.3%-13% mis-labeled images in the original CASIA-WebFace, so we can regard WebFace-Clean as a noise-free dataset. Several experiments are performed, and their input and training settings are described in the Section 5.

We firstly build another new noise-free FR dataset, named **WebFace-Clean-Sub**, which contains 60% images randomly chosen from all celebrities in WebFace-Clean. The remaining 40% images are used to synthesis noisy data, named **WebFace-Noisy-Sub**. Noise in FR datasets mainly fall into two types: label flips, where an image has been given a label of another class within the dataset, and outliers, where an image does not belong to any of the classes, but mistakenly has one of their labels, or non-faces can be found in the image. To generate Web-Noisy-Sub, we synthesize label flips noise by randomly changing face labels into incorrect classes, and simulate outlier noise by randomly polluting data with images from MegaFace [19], and we keep the ratio of **label flips** and **outliers** at 50%:50%. Therefore, we get a new noisy dataset **WebFace-All** containing WebFace-Clean-Sub and WebFace-Noisy-Sub, and its noise rate is 40%.

A ResNet-20 model(CNN-All-L2) [24] supervised with L2-Softmax($s = 32$) is trained with WebFace-All. For comparison, we also train another ResNet-20 model(CNN-Clean-L2) with WebFace-All, but with a small modification: a sample will be dropped in training if it belongs to WebFace-Noisy-Sub. Therefore, CNN-Clean-L2 can be considered to be trained only with clean samples.

In each training iteration, we use $\mathbf{W}_j$ to compute $\theta_{i,j}$ for all samples belonging to the j-th class. For simplification, only $cos\theta$ is computed in our implementation. We refer the $\cos\theta$ histogram distributions(the bin size is 0.01) of all training samples as $Hist_{all}$. Moreover, we compute the distributions of clean and noisy samples separately as $Hist_{clean}$ and $Hist_{noisy}$.

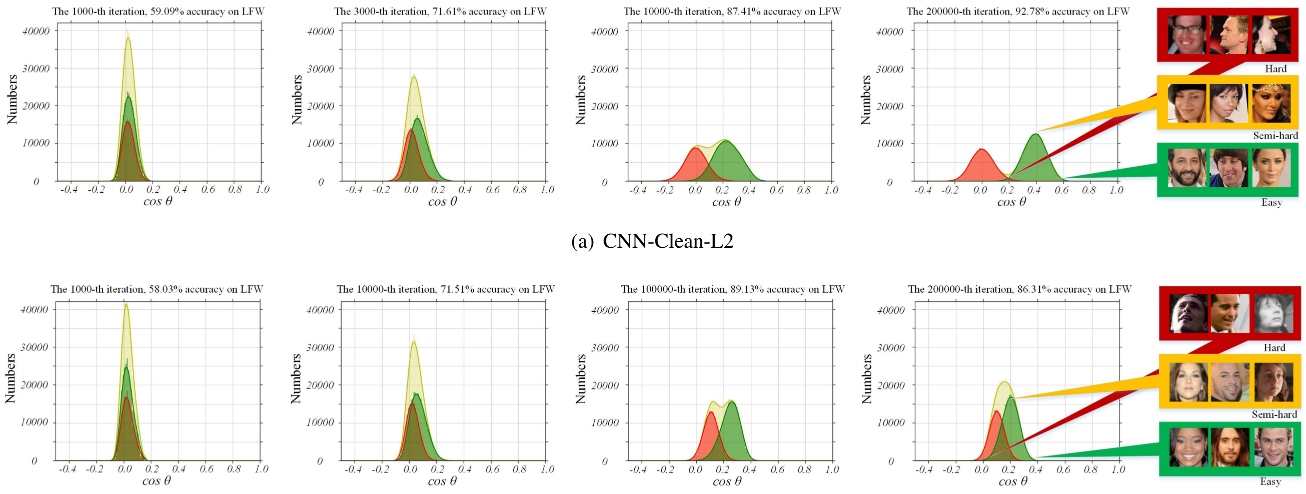
Figure 1 shows the distributions of CNN-Clean-L2 and CNN-All-L2. The test accuracy on LFW [16] is used to demonstrate the performance of the trained CNNs. From Figure 1, we have following observations:

1. $Hist_{clean}$ and $Hist_{noisy}$ are all Gaussian-like distributions throughout the training process for CNN-All-L2 and CNN-Clean-L2. Experiments in ArcFace [5] also demonstrate this phenomenon. The Gaussian-like distributions should be caused by similar quality distributions [42] in FR datasets.

2. At the beginning of training,$Hist_{clean}$ and $Hist_{noisy}$ are largely overlapping, making them impossible to be separated from each other, since the CNNs are initially untrained.

3. After a short period, $Hist_{clean}$ starts to move to the right. If noisy samples are involved in training (CNN-All-L2), $Hist_{noisy}$ moves to the right too, but it has been always on the left side of $Hist_{clean}$. Therefore, the samples with larger $cos\theta$ values have larger probability to be clean. This phenomenon is mainly because that the CNNs memorize easy/clean samples quickly, and can also memorize hard/noisy samples eventually [1].

[1] github.com/happynear/FaceVerification



(a) CNN-Clean-L2

(b) CNN-All-L2

Figure 1. The $cos\theta$ histogram distributions of CNN-Clean-L2 (top) and CNN-All-L2 (bottom). $Hist_{clean}$, $Hist_{noisy}$ and $Hist_{all}$ are colored with Green, Red and Yellow respectively. The curve edge of each distribution is smoothed with a mean filter with size = 5 to remove noise (described in the Section 4.2.1).

4. In the latter stage of training process, the $cos\theta$ of a sample in $Hist_{clean}$ reflects the quality of the corresponding face image. Some face images of easy, semi-hard, and hard clean samples are provided in Figure 1.

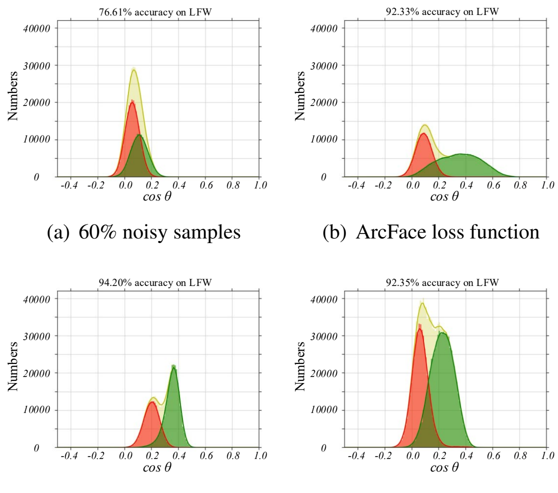
5. The performance of CNN-All-L2 is adversely affected by noisy data. From the distribution in Figure 1(b), we can observe the negative impact in two aspects: (1) $Hist_{clean}$ and $Hist_{noisy}$ have large overlapping regions throughout the training process; (2) Compared with the $Hist_{clean}$ in Figure 1(a), the $Hist_{clean}$ in Figure 1(b) is on the left side.

These observations can be explained in theory, and more experiments are further performed to confirm them: (1)We increase the noise rate from 40% to 60%; (2)ArcFace [5] is employed to supervise the CNNs; (3) we replace the ResNet-20 with a deeper ResNet-64 [24]; (4) Another clean dataset IMDB-Face [42][2] is chosen to replace WebFace-Clean. Their final $cos\theta$ distributions shown in Figure 2 further approve our observations.

CNN-Clean-L2 is actually trained by using a ideal paradigm: ignoring all noisy sample in training. However, it is difficult to predict if a sample is noisy in real training. In this paper, we propose a paradigm to minimize the impact of noisy samples based on the $cos\theta$ distributions.

4. The Proposed Paradigm

We propose a new training paradigm for learning face features from large-scale noisy data based on above observations. In each mini-batch training, we compute $cos\theta$ for all training samples, and the current distribution $Hist_{all}$. We define δ_l and δ_r are the leftmost/rightmost $cos\theta$ values in $Hist_{all}$. Based on the first observation in Section 3.2, no more than 2 peaks can be detected in $Hist_{all}$. μ_l and μ_r denote the $cos\theta$ values of the left/right peaks respectively, and $\mu_l = \mu_r$ if there is only one peak.



(a) 60% noisy samples

(b) ArcFace loss function

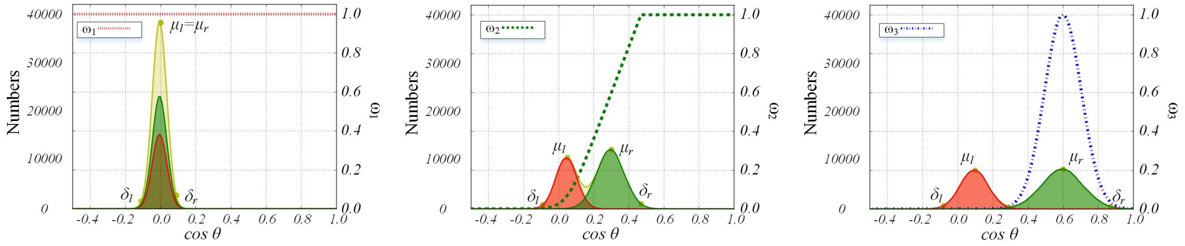
(c) ResNet-64 model

(d) IMDB-Face dataset

Figure 2. The final $cos\theta$ distributions of other four models. These distributions further confirm our observations.

The target of the paradigm is to correctly evaluate the probability of a sample being clean in training, and then adjust the weight of the sample according to this probability. The key idea of our paradigm can be briefly introduced as:

[2] We only downloaded 1.2M images of total 1.7M images with the provided URLs



(a) All samples are equally treated (b) Clean samples are emphasized (c) Semi-hard clean samples are emphasized

Figure 3. The $cos\theta$ distributions and the corresponding ω of three strategies.

1. At the beginning of training process, all samples are treated equally.

   According to the 2nd observation in Section 3.2, at the beginning of training process, the trained CNN does not have the ability for face recognition. Therefore, all samples should have the same weight for training.

2. After a short period of training, samples with larger $cos\theta$ have larger weight.

   According to the 3rd observation in Section 3.2, after a short period of training, samples with larger $cos\theta$ should have larger weight for training.

3. In the end of training, semi-hard clean samples are emphasized to further promote the performance.

   According to the 4th observation in Section 3.2, easy, semi-hard and hard clean samples can be distinguished according to their $cos\theta$ in $Hist_{clean}$ at this time. The pre-condition of training semi-hard clean samples is that the trained CNN already has good performance, and the overlapping area of $Hist_{clean}$ and $Hist_{noisy}$ is relatively small. In this circumstance, semi-hard clean samples will have larger weight than other samples for training. Training semi-hard clean samples is also implicitly adopted in ArcFace [5] and FaceNet [36].

We present three corresponding strategies to compute sample weights as following:

Strategy One In this strategy (see Figure 3(a)), all samples have the same weight as

$$\omega_{1,i} = 1, \tag{4}$$

where $\omega_{1,i}$ is the weight of an input sample $\mathbf{x}_i$.

Strategy Two In this strategy (see Figure 3(b)), the samples with larger $cos\theta$ have larger weight as

$$\omega_{2,i} = \frac{softplus(\lambda z)}{softplus(\lambda)}, \tag{5}$$

where $z = \frac{cos\theta_{i,j} - \mu_l}{\delta_r - \mu_l}$, and $softplus(x) = log(1 + e^x)$ is a smooth version of the RELU activation[29]. λ is used to normalize the function, and $\lambda = 10$ in all experiments.

Strategy Three In this strategy (see Figure 3(c)), the semi-hard clean samples are emphasized. We define μ_r as the $cos\theta$ value of the right peak in $Hist_{all}$ (μ_l corresponding to the left peak), which can be consider as the center of $Hist_{clean}$ (according to the 5-th and the 2-th observation). The sample weight is computed as

$$\omega_{3,i} = e^{-(cos\theta_{i,j} - \mu_r)^2 / 2\sigma^2}, \tag{6}$$

where σ^2 is the variance of the $Hist_{clean}$, which can be approximated by using the part to the right of μ_r. We set $\sigma = (\delta_r - \mu_r)/2.576$ to cover 99% samples in $Hist_{clean}$.

4.1. Compute Time-Varying Fusion Weight

Three weighting sample strategies are introduced as above, but how to select the applied strategy in training? There is no clear criteria. Inspired by the gradually learning technique in Co-teaching [13], we compute the sample weight in a fusion way.

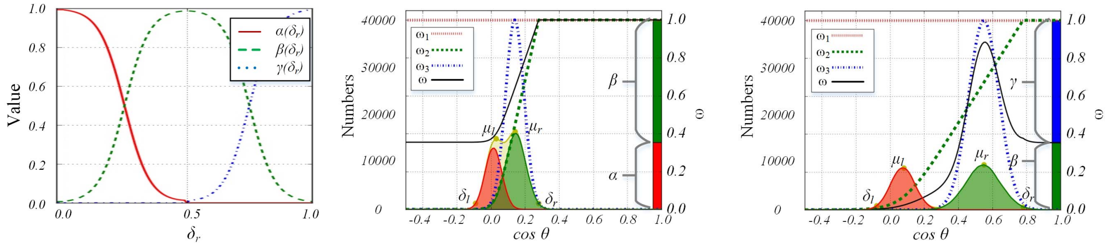
According to our observations, δ_r is a good signal to approximately reflect the performance of the trained CNN. As the CNN achieves better performance, δ_r will gradually move to the right. We define a threshold ζ to divide possible δ_r values into two ranges: $[0, \zeta]$ and $(\zeta, 1]$. The selected strategy is changed from the 1st one to the 2nd one in the first range, and from 2nd one to the 3rd one in the second range. In all experiments, we set $\zeta = 0.5$. Then, we compute the sample weight as

$$\omega_i = \alpha(\delta_r)\omega_{1,i} + \beta(\delta_r)\omega_{2,i} + \gamma(\delta_r)\omega_{3,i}, \tag{7}$$

where

$$\alpha(\delta_r) = (2 - \frac{1}{1 + e^{5 - 20\delta_r}} - \frac{1}{1 + e^{20\delta_r - 15}})\lceil 0.5 - \delta_r \rceil, \tag{8}$$

, $\beta(\delta_r) = 1 - \alpha(\delta_r) - \gamma(\delta_r)$ and $\gamma(\delta_r) = \alpha(1.0 - \delta_r)$. As shown in Figure 4, at first, $\omega_{1,i}$ has the greatest impact. As δ_r gradually moves to right, $\omega_{2,i}$ and then $\omega_{3,i}$ begin to play more important roles.



(a) $\alpha(\delta_r)$, $\beta(\delta_r)$, and $\gamma(\delta_r)$

(b) Two fusion examples

Figure 4. The left figure demonstrates three functions: $\alpha(\delta_r)$, $\beta(\delta_r)$, and $\gamma(\delta_r)$. The right figure shows two fusion examples. According to the ω, we can see that the easy/clean samples are emphasized in the first example($\delta_r < 0.5$), and the semi-hard clean samples are emphasized in the second example($\delta_r > 0.5$).

4.1.1 Discussion on Weight Computation

Equation 4, Equation 5 and Equation 6 are used to compute the sample weight during training. These 3 equations directly reflect the key idea of our paradigm, but they are only introduced empirically and have some free parameters. The same situation also exists in Equation 7. The key contribution of our approach is the ideas of training paradigm and weight fusion. We also perform some experiments, and found that good performance can be achieved as long as the used equations can correctly reflect the key ideas. Therefore, these equations are provided for reference, and more exploration could be conducted to find other theoretical justified equations.

4.2. Implementation Details

4.2.1 $Hist_{all}$ Related Variables

According to the Equation 7, $cos\theta_{i,j}$, δ_r, δ_l and μ_r are required to compute the final weight of the sample $\mathbf{x}_i$ belonging to the j-th class. Except $cos\theta_{i,j}$, other variables are computed based on $Hist_{all}$.

In theory, $Hist_{all}$ should be computed in each mini-batch training, which is very time-consuming because the number of samples is usually very large in FR datasets. In our implementation, the $cos\theta$ values of recent K training samples are stored to compute another distribution $Hist_K$. The training samples are pre-shuffled, $Hist_K$ can be considered as an approximate $Hist_{all}$ with a suitable K. We set $K = 64,000$ (1000 batches) in our experiments.

To resist noise, a mean filter with size 5 is firstly applied to $Hist_K$ to remove noise. We select the top 0.5% leftmost/rightmost $cos\theta$ values as δ_l and δ_r. A very simple method is applied to find all peaks in $Hist_K$: the number of frequency in a bin is larger than all of its left/right neighbour bins (Radius = 5). Theoretically, we can only find one or two peaks during training. However, we sometimes find more than two peaks, or find no peak at all, because $Hist_{clean}$ and $Hist_{noisy}$ are actually not always Gaussian-like distributions. We employ a simple technique to find μ_r: if there is only one peak $\in (\zeta, 1]$, its $cos\theta$ is the μ_r, and if more than one peaks are found, we choose the highest one. Similar method can be used to find μ_l.

When the noise rate is very high, μ_r may become difficult to detect. In this circumstance, the key is to train easy/clean samples as much as possible, and $\omega_{2,i}$ should play more important role than $\omega_{3,i}$. Therefore, missing μ_r may have few impact on the final performance. In the contrary, if the noise rate is very low, missing μ_l may also have few impact on the final performance.

4.2.2 Weighting in AM-Losses

Method 1 Usually, the weight is applied to minimize the loss and the weighted loss function of L2-Softmax is

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\omega_i log\frac{e^{s\ cos\ \theta_{i,y_i}}}{\sum_{j=1}^{C} e^{s\ cos\ \theta_{i,j}}}. \tag{9}$$

Moreover, there is another method to apply sample weights.

Method 2 In AM-Losses, an input $\mathbf{x}_i$ is normalized and re-scaled with a parameter s ($\|\mathbf{x}_i\|$ can be regarded as s in SphereFace). The scaling parameter s is better to be a properly large value as the hypersphere radius, according to the discussion in [45, 5]. A small s can lead to insufficient convergence even no convergence. The lower bound of s is also discussed in [45, 33]. Inspired by the effect of s, we can also apply the weights ω_i to the scaling parameter during training CNNs. Therefore, the loss function of L2-Softmax can be formulated as

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{\omega_i s\ cos\ \theta_{i,y_i}}}{\sum_{j=1}^{C} e^{\omega_i s\ cos\ \theta_{i,j}}}. \tag{10}$$

Similar method can be applied in other AM-Losses too. Two methods all can be employed to train CNNs with noisy datasets. According to our experiments, the latter one shows better performance in most cases.

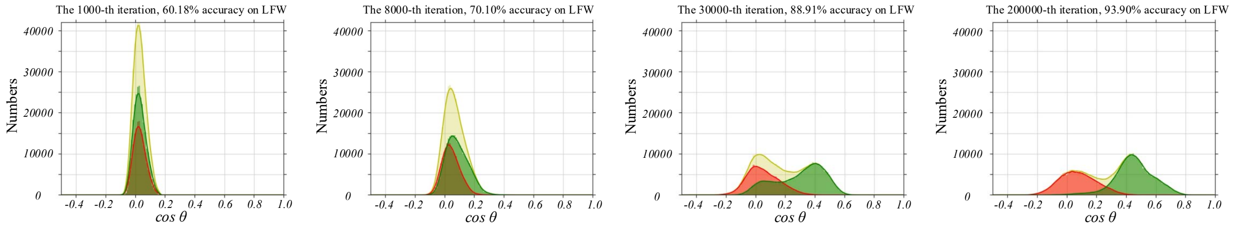


Figure 5. The $cos\theta$ distributions of our CNN model trained with 40% noisy samples.

5. Experiments

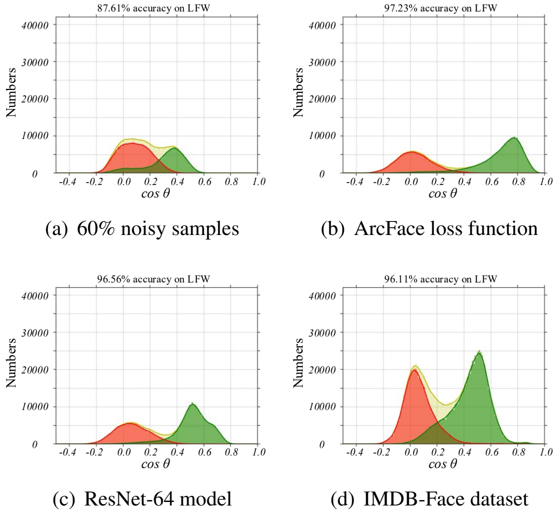
To verify the effectiveness of our method, several experiments are performed. In these experiments, face images and landmarks are detected by MTCNN [52], then aligned by similar transformation as [49], and cropped to 128×128 RGB images. Each pixel in RGB images is normalized by subtracting 127.5 then dividing by 128. We use Caffe [17] to implement CNN models. For fair comparison, all CNN models are trained with SGD algorithm with the batch size of 64 on 1 TitanX GPUs. The weight decay is set to 0.00005. The learning rate is initially 0.1 and divided by 10 at the 80K, 160K iterations, and we finish the training process at 200K iterations.

First, we perform a similar experiment as in the Section 3.2, but using the proposed training paradigm. Figure 5 shows the $cos\theta$ distributions during training. It is obvious that $Hist_{clean}$ are separated from $Hist_{noisy}$. According to the final distributions in Figure 5, Figure 1(a) and Figure 1(b), the adverse effect from noisy samples is largely eliminated this time.

Corresponding to 4 models in Figure 2, we re-train them using the proposed paradigm, and the final distributions are shown in Figure 6. Our method also gets better results.

We perform experiments with different noise rates, supervised AM-Losses and computing weighted loss methods(see Section 4.2.2) with the experiment in the Section 3.2. The models are evaluated on Labelled Faces in the Wild (LFW) [16], Celebrities in Frontal Profile (CFP) [37], and Age Database (AgeDB) [28]. As shown in Table 1, competitive performance can be achieved using our paradigm, without any prior knowledge about noise in training data. We can surprisedly see that some results of CNN_{m2} are even better than the results of CNN_{clean}. This improvement is mainly caused by semi-hard training in the final stage. It can be seen that the 2nd method in Section 4.2.2 demonstrates a better performance.

For comparison, we also implemented a recently proposed noise-robust method for image classification: Co-teaching [13] (CNN_{ct}), which selects small-loss samples from each mini-batch. **Note** the noise rate should be pre-given in Co-teaching. The results prove that general noise-robust approaches cannot achieve satisfied performance in FR.



(a) 60% noisy samples

(b) ArcFace loss function

(c) ResNet-64 model

(d) IMDB-Face dataset

Figure 6. The final $cos\theta$ distributions of four models (corresponding to four models Figure 2) using our paradigm.

5.1. Estimating Noise Rate

There is an interesting observation from Figure 5 and Figure 6: at the end of training process, the region on the left of μ_l approximately contains half of noisy samples, so we can estimate the noise rate in the training dataset. If the left peak (μ_l) is not detected, the region on the right of μ_r, which contains about half of clean samples, also can be used. The estimated rates in Table 1 further prove the effectiveness of our method.

5.2. Learning from Original MS-Celeb-1M

The original MS-Celeb-1M [11] contains 99,892 celebrities, and 8,456,240 images. For comparison, two ResNet-64 [24], CNN_{ours} and CNN_{normal}, supervised with ArcFace [5] are employed to learn face features from MS-Celeb-1M, one using the proposed paradigm and the other not. To accelerate convergence speed, these ResNet-64 are firstly trained with Casia-WebFace [51], then finetuned with MS-Celeb-1M. Other training parameters are similar with the previous experiments. A refined MS-Celeb-1M, containing 79,077 celebrities and 4,086,798 images, is provid-

Loss	Noise	CNN_{clean}			CNN_{normal}			CNN_{ct}	CNN_{m1}	CNN_{m2}			Estimated
	Rate	LFW	AgeDB	CFP	LFW	AgeDB	CFP	LFW	LFW	LFW	AgeDB	CFP	Noise Rate
L2-Softmax	0%	94.65	79.95	82.04	94.65	79.95	82.04	-	95.00	**96.28**	**84.05**	**87.88**	2%
	20%	94.18	79.33	81.00	89.05	66.83	71.55	92.12	92.95	**95.26**	**81.91**	**84.77**	18%
	40%	92.71	76.51	77.10	85.63	58.95	68.78	87.10	89.91	**93.90**	**78.38**	**81.37**	42%
	60%	**91.15**	**70.28**	**74.74**	76.61	51.38	63.12	83.66	86.11	87.61	64.43	70.54	56%
ArcFace	0%	97.95	88.48	**91.07**	97.95	88.48	91.07	-	97.11	**98.11**	**88.61**	90.81	2%
	20%	**97.80**	**88.75**	89.54	96.48	82.83	82.52	96.53	96.83	97.76	88.46	**90.22**	18%
	40%	96.53	84.93	84.81	92.33	72.68	74.11	94.25	95.88	**97.23**	**86.03**	**88.41**	36%
	60%	94.56	80.75	80.52	84.05	58.73	67.70	90.36	93.66	**95.15**	**81.45**	**83.25**	54%

Table 1. Comparison of accuracies(%) on LFW, AgeDB(30), and CFP(FP). ResNet-20 models are used. CNN_{clean} is trained only with clean data (WebFace-Clean-Sub) as *Upper Bound*. CNN_{normal} is trained with the noisy dataset WebFace-All using the traditional method. CNN_{ct} is trained with WebFace-All using our implemented Co-teaching(with pre-given noise rates). CNN_{m1} and CNN_{m2} are all trained with WebFace-All but using the proposed approach, and they respectively use the 1st and 2nd method to compute loss(see Section 4.2.2). CNN_{m1} and CNN_{ct} are only evaluated on LFW.

ed in LightCNN [49], so the noise rate is about 51.6%. We also train a CNN (CNN_{clean}) with the refined MS-Celeb-1M for comparison. $Hist_{all}$, together with $Hist_{clean}$ and $Hist_{noisy}$ approximated according to the refined dataset, are presented in Figure 7. According to $Hist_{noisy}$, we estimate that the noise rate of the original MS-Celeb-1M is about 43%.

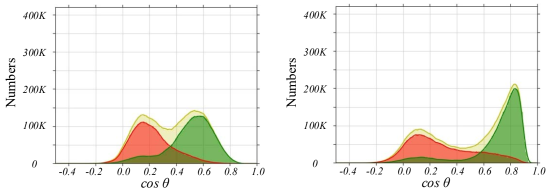


Figure 7. The final $cos\theta$ distributions of CNN_{normal} (left) and CNN_{ours} (right).

The trained CNNs are then evaluated on LFW, AgeDB-30, CFP-FP, YTF [48] and MegaFace Challenge 1 [19], as shown in Table 2. CosFace [45] and ArcFace [5] are added for comparison since they use the same network or AM-Loss with ours. The competitive performance of CNN_{ours} demonstrates the effectiveness of our training paradigm.

6. Conclusion and Future Work

In this paper, we propose a FR training paradigm, which employs the idea of weighting training samples, to train AM-Loss supervised CNNs with large-scale noisy data. At different stages of training process, our paradigm adjusts the weight of a sample based on the $cos\theta$ distribution to improve the robustness of the trained CNN models. Experiments demonstrate the effectiveness of our approach. Without any prior knowledge of noise, the CNN model can be directly trained with an extremely noisy dataset ($> 50\%$ noisy samples), and achieves comparable performance with the model trained with an equal-size clean dataset. Moreover, the noise rate of a FR dataset can also be approximated with our approach.

Method	LFW	AgeDB	CFP	YTF	MF1
CosFace [45]	99.73	-	-	97.6	77.11
ArcFace [5]	99.83	98.08	96.82	-	83.27
CNN_{normal}	99.21	90.85	93.38	95.64	70.16
CNN_{clean}	99.67	96.50	95.74	97.12	78.21
CNN_{ours}	99.72	96.70	96.40	97.36	78.69

Table 2. Comparison of accuracies(%) on several public benchmarks. Accuracies of CosFace and ArcFace are cited from their original papers. CosFace is trained with a clean dataset containing 90K identities and 5M images. ArcFace is trained with a manually refined dataset containing 93K identities and 6.9M images. Their datasets all are composed of several public datasets including refined VGG2 [3], MS-Celeb-1M, etc. CNN_{ours} and CNN_{normal} are trained only with the original noisy MS-Celeb-1M(**noise rate** $\approx$ 50%). CNN_{clean} is trained with the refined MS-Celeb-1M [49]. An improved ResNet-100 is used in ArcFace, and other 4 methods all use a ResNet-64 CNN model.

The proposed paradigm also has its limitations. Firstly, it shares the same limitations with most of noise-robust training methods: the hard clean samples also have small weight, which might affect the performance. However, reducing effects of noisy samples should have higher priority while learning with a heavy noisy dataset. Secondly, Guassian-like distributions cannot be guaranteed throughout the whole training process. Fortunately, our method to find left/right peaks and end points does not heavily depend on this assumption. Lastly, we need to study the reason that the 2nd method is superior to the 1st method in Section 4.2.2.

To conclude, this work will greatly reduce the requirement for clean datasets when training FR CNN models, and makes constructing huge-scale noisy FR datasets a valuable job. Moreover, our approach also can be employed to help refine a large-scale noisy FR dataset.

References

[1] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *IMCL*, 2017. 3

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 2

[3] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 67–74. IEEE, 2018. 8

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546. IEEE Computer Society, 2005. 2

[5] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. June 2018. 1, 2, 3, 4, 5, 6, 7, 8

[6] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2006–2014, 2017. 2

[7] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014. 2

[8] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001. 2

[9] A. Ghosh, H. Kumar, and P. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, pages 1919–1925, 2017. 1, 2

[10] J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017. 1, 2

[11] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102, 2016. 1, 7

[12] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama. Masking: A new perspective of noisy supervision. In *NIPS*, 2018. 1, 2

[13] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NIPS*, 2018. 1, 2, 5, 7

[14] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NIPS*, 2018. 2

[15] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In A. Feragen, M. Pelillo, and M. Loog, editors, *SIMBAD*, volume 9370 of *Lecture Notes in Computer Science*, pages 84–92. Springer, 2015. 2

[16] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. *Technical Report, University of Massachusetts*, 2007. 3, 7

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. 7

[18] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2309–2318, 2018. 1, 2

[19] I. Kemelmachershlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 1, 3, 8

[20] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010. 2

[21] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936, 2017. 1, 2

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[23] T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016. 2

[24] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 6738–6746. IEEE Computer Society, 2017. 1, 2, 3, 4, 7

[25] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on International Conference on Machine Learning*, pages 507–516, 2016. 2

[26] E. Malach and S. Shalev-Shwartz. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017. 1, 2

[27] A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning*, pages 125–134, 2015. 2

[28] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1997–2005, 2017. 7

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 5

[30] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013. 2

[31] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2233–2241, 2017. 1, 2

[32] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 3

[33] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv:1703.09507*, 2017. 6

[34] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, 2018. 2

[35] D. Rolnick, A. Veit, S. Belongie, and N. Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017. 1

[36] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823. IEEE Computer Society, 2015. 2, 5

[37] S. Sengupta, J. C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9, 2016. 7

[38] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. In *ICLR*, 2015. 1, 2

[39] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10, 000 classes. In *CVPR*, pages 1891–1898. IEEE Computer Society, 2014. 2

[40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708. IEEE Computer Society, 2014. 2

[41] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 6575–6583, 2017. 1, 2

[42] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy. The devil of face recognition is in the noise. In *European Conference on Computer Vision*, pages 780–795. Springer, 2018. 1, 3, 4

[43] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 1, 2

[44] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM Conference on Multimedia*, 2017. 2

[45] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. June 2018. 1, 2, 6, 8

[46] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *ECCV (7)*, volume 9911 of *Lecture Notes in Computer Science*, pages 499–515. Springer, 2016. 2

[47] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515, 2016. 2

[48] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 529–534, Washington, DC, USA, 2011. IEEE Computer Society. 8

[49] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, PP(99):1–1, 2015. 1, 7, 8

[50] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015. 1, 2

[51] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 2, 3, 7

[52] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 7

[53] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tailed training data. In *IEEE International Conference on Computer Vision*, pages 5419–5428, 2017. 2

[54] Y. Zheng, D. K. Pal, and M. Savvides. Ring loss: Convex feature normalization for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2