Unequal-training for Deep Face Recognition with Long-tailed Noisy Data

Yaoyao Zhong, Weihong Deng[*], Mei Wang, Jiani Hu
Beijing University of Posts and Telecommunications
{zhongyaoyao, whdeng, wangmei1, jnhu}@bupt.edu.cn

Jianteng Peng, Xunqiang Tao, Yaohai Huang
Canon Information Technology (Beijing) Co., Ltd
{pengjianteng, taoxunqiang, huangyaohai}@canon-ib.com.cn

Abstract

Large-scale face datasets usually exhibit a massive number of classes, a long-tailed distribution, and severe label noise, which undoubtedly aggravate the difficulty of training. In this paper, we propose a training strategy that treats the head data and the tail data in an unequal way, accompanying with noise-robust loss functions, to take full advantage of their respective characteristics. Specifically, the unequal-training framework provides two training data streams: the first stream applies the head data to learn discriminative face representation supervised by Noise Resistance loss; the second stream applies the tail data to learn auxiliary information by gradually mining the stable discriminative information from confusing tail classes. Consequently, both training streams offer complementary information to deep feature learning. Extensive experiments have demonstrated the effectiveness of the new unequal-training framework and loss functions. Better yet, our method could save a significant amount of GPU memory. With our method, we achieve the best result on MegaFace Challenge 2 (MF2) given a large-scale noisy training data set.

1. Introduction

Deep convolutional neuron networks (DCNNs) have achieved great success in computer vision [11,12,18,19,29], significantly improving the state of art in face recognition [6, 7, 21, 28, 36, 37, 39]. Besides the evolving architectures, large-scale training datasets play a crucial role in deep face recognition. It is worth to point out that real-world face datasets are usually large-scale and exhibit a long-tailed distribution, which present three challenges for model training. First, there exists extremely imbalanced identities in such a large dataset, in which some identities have sufficient samples, while for other hundreds of thousands identities, only very few samples are available. Second, there is significant noise inherent in the long-tailed face datasets. As reported in [35], the label noise percentage increases dramatically as the scale of data growing, and million level datasets typically would even have a noise ratio higher than 30%. Third, with a massive number of identities, the fully-connected layer connected with softmax loss will become extremely large, thus the GPU memory will be congested and the batch-size will be lowered, which will make the training loss difficult to converge [34].

The three challenges, *i.e.* extremely unbalanced data, million level identities, and inherent noise, undoubtedly aggravate the difficulty of training. The experiment of Zhang *et al.* [44] indicates that, a model trained on the whole long-tailed dataset will perform worse than that trained on a specific proportion of the whole dataset (cutting 50% tail in their work). The phenomenon indicates that, it would be sub-optimal to train on the whole face dataset without considering characteristics of the data. The tail identities cannot provide an accurate description with limited number of training samples, thus the feature space of them will be squeezed by the head identities. Moreover, overfitting to noise would further deteriorate the model [35].

Unfortunately, current training methods cannot stably make full use of discriminative information in the long-tailed noisy dataset. Due to this complicated situation of long-tailed noisy face dataset, traditional methods, *e.g.* re-sampling [3] and cost-sensitive weighting [17], are no longer feasible. Some recently proposed solutions attempted to alleviate long-tailed problem by compensating the tail data [41, 43, 44]. Although they can treat the head and tail data equally, these methods may by easily affected by the label noise. Thus, we dedicate to tackling the long-tailed problem in deep face recognition, improving the resistance of training models to noise, exploring effective

[*] Corresponding Author

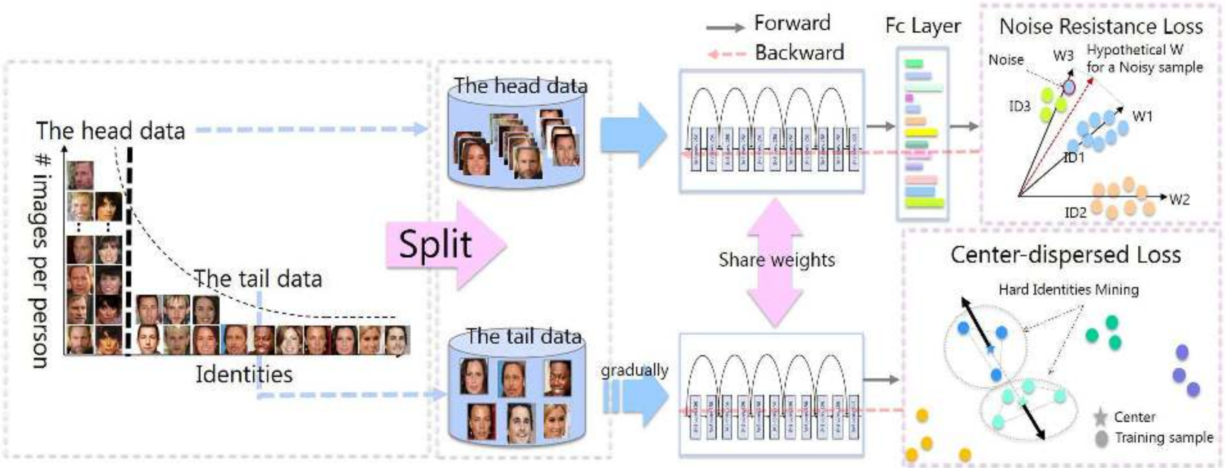

Figure 1. The unequal-training framework based on the long-tailed dataset and the corresponding loss functions, provides two training data streams to the model: the first stream, based on the head data, is used for training relative discriminative face representation supervised by a noise-resistant loss which incorporates a hypothetical training face representation in feature space; the second stream, based on the tail data, is used to learn the stable inter-class discriminative information by mining hard identities, adding them gradually in an iterative way and enhancing the feature space by dispersing them.

method to use the accurate information as much as possible to strengthen the model.

In this paper, we propose an unequal training method to treat the head data and the tail data differently, which can take full advantage of discriminative information stably. The head identities with abundant samples are suitable to characterize the intra-class variability while the large number of tail identities could offer abundant inter-class information. Specifically, the unequal-training strategy based on the long-tailed dataset and the corresponding loss functions, provides two training data streams to the model: (1) the first stream, based on the head data, is used for training relative discriminative face representation supervised by a noise-resistant loss; (2) the second stream, based on the tail data, is used to learn the stable inter-class discriminative information by hard identities mining. Based on the initial model of the first stream, the mining procedure gradually enhances model stably by iteratively mining the most valuable inter-class information. The main contributions of this paper can be summarized as follows:

1. We delve into the long-tailed noisy dataset, and propose an two-stream unequal-training framework that deals with the head data and the tail data differently. To the best of our knowledge, this is the first work in the deep face recognition literature to deal with different parts of long-tailed noisy data separately according to the identity distribution. Besides, our framework also saves a large proportion of GPU memory compared to the classical cross-entropy softmax loss.

2. We analyze the characteristics of the label noise in the long-tailed face dataset, and propose the corresponding loss functions to deal with the noise in the head and tail data respectively. For the noisy tail data, we propose an iterative way to gradually train the tail data, in which the hard identities mining makes sure the most stable information could be preserved.

3. Extensive experiments on CASIS-Webface [42], MegaFace Challenge 2 (MF2) [24], LFW [15], Cross-Pose LFW (CPLFW) [45] and [40] datasets have demonstrated the effectiveness of our unequal-training framework and new loss functions. In particular, our method achieves the state-of-the-art result on MegaFace Challenge 2 (MF2) [24] given a large-scale noisy training data set.

2. Related Work

Deep learning has brought great success to face recognition recently and the major focus in face recognition has become to learn a discriminative feature space by supervising networks using effective loss.

The Contrastive loss [6], Triplet loss [28] and Quintuplet loss [13] learn feature representations using pair samples. This type of loss functions get rid of supervision of softmax loss so that it could save GPU memory in large-scale training. However, they may suffer from time-consuming mining of hard examples and this circumstance often occurs when the training data expand dramatically. Another euclidean metric learning methods are based on classification, such as Centerloss [39], Rangeloss [44], and Marginal loss [8]. They usually serve as an auxiliary loss for softmax loss aiming at learning a more discriminative feature space. A more powerful type of loss function is large margin softmax loss, mainly containing SphereFace [21](L-Softmax [22]), CosFace [36] and ArcFace [7], which significantly boost the face recognition.

However, both classification-based euclidean metric learning and Large margin softmax prefer more uniform and sufficient training data. Besides, when enlarging the training identities to million level, the GPU memory becomes congested and the batch size is lowered.

The long-tailed problem in face recognition is reminiscent of the conventional class imbalance problem that has been comprehensively studied in classical machine learning [2, 10], but significantly differs from conventional class imbalance problem in two aspects: first, the long-tailed data in face recognition is large-scale, with millions of identities; second, the long-tailed data is inherently noisy. Thus, traditional methods such as data re-sampling [3] and cost-sensitive weighting [17] are no longer feasible here.

There are only a few works making preliminary attempts to investigate the long-tailed effect in deep face recognition. Quintuplet loss [13] reduces the class imbalance inherent in the local data neighborhood, enforcing both inter-cluster and inter-class margins. Rangeloss [44] reduces overall intra-personal differences and enlarges inter-personal differences in one mini-batch, promoting tail data by local refinement. Center invariant loss [41] balances feature spaces of different training identities by aligning the centers of each identity. A feature transfer approach [43] proposed by Xi *et al*. also promotes the tail to balance training data, by generating feature-level samples through transfer of intra-class variance from head data.

Previous works discover that feature space of tail identities is squeezed by head identities. Worse still, bad spitting of the feature space leads to a bad generalization ability. Researchers have racked brains trying to promote the tail data so that all the data are treated equally. Although they can treat the head and tail data equally, these methods may by easily affected by the label noise. Furthermore, noisy training samples of a tail identity could bring more risk compared with those of a head identity at the same level noise, which reflects that tail identities and head identities may be affected differently by noise.

Therefore we argue that whether we should expect that all the data contribute equally to the feature space. Why not leverage identities with different samples respectively according to their characteristics? Wang *et al*. [38] propose to leverage the head tail to build a reliable model, then use the information from the tail in an unsupervised way to improve the robustness of the original model, which is a similar work to us. The difference is that our method concentrates more on the difference of the two part data while they focus on leveraging the general knowledge of them.

3. The Approach

We first provide an overview of the proposed unequal-training framework, shown in Figure 1. Our approach consists of three steps:

(1) **Splitting the training dataset -** Given a long-tailed face training dataset, we split the dataset into the head data and the tail data according to the distribution. The head data is defined as the largest portion of majority identities, on which we could train the model using softmax-based loss better than on other portions and even on the whole long-tailed dataset. Correspondingly, the rest of the long-tailed dataset is defined as the tail data. The head data and the tail data will provide two training data streams to the model.

(2) **Constructing a noise resistance model -** Training with the whole long-tailed data will deteriorate the model inevitably. It is a reasonable compromise between abundance and balance of identities to learn relative discriminative face representation using the head data which could characterize the intra-class variability. Relatively discriminative face representations are learned on the head data supervised by a noise resistance loss which incorporates a hypothetical training face representation in feature space.

(3) **Joint training with the tail data -** Finally, we re-train the model with two stream data: the first stream, based on the head data, is used for stabilizing the face representation supervised by a noise-resistant loss; the second stream, based on the tail data, is used for enhancing the model by learning the stable inter-class discriminative information. We mine hard identities, add them gradually in an iterative way and enhance the feature space by dispersing them.

3.1. Constructing a noise resistance model

In this section, we introduce the detailed process of learning relative discriminative face representation using the head data.

The head data is relatively abundant and balanced so that people will naturally think that softmax-based loss, including classical softmax and large margin softmax, could be employed to train a base model. softmax-based loss is effective, yet quite remarkably, it could be deteriorated severely by training with contaminated dataset according to recent research [35]. Considering the existence of considerable and evenly distributed noise in the head data, we have to enhance the robustness to the noise of softmax-based loss.

First we analyze the types of noise in the training set, and the differences between noisy training data and correctly labeled data, so that we could distinguish noisy data from clean ones. There are mainly three types of noise shown in Figure 2 in the face traning dataset. (1) Label flips, where the image has been incorrectly labeled as another identity of the training dataset. (2) Outliers, where the image has been incorrectly labeled as the identity i of the training dataset. The image actually does not belong to any identities of the training dataset, but it is highly similar-looking as another identity j of the training dataset so that in the training process it is predicted by the model as identity j. (3) Entirely dirty data, where the image has been incorrectly labeled as an identity of the training dataset. But the image actually does not belong to any of the identities of the training dataset as the second type noise. It even does not belong to any identities in face recognition. The difference between entirely dirty data and the second type outliers is that entirely dirty data could not be classified as any identities of the training dataset in the training process.

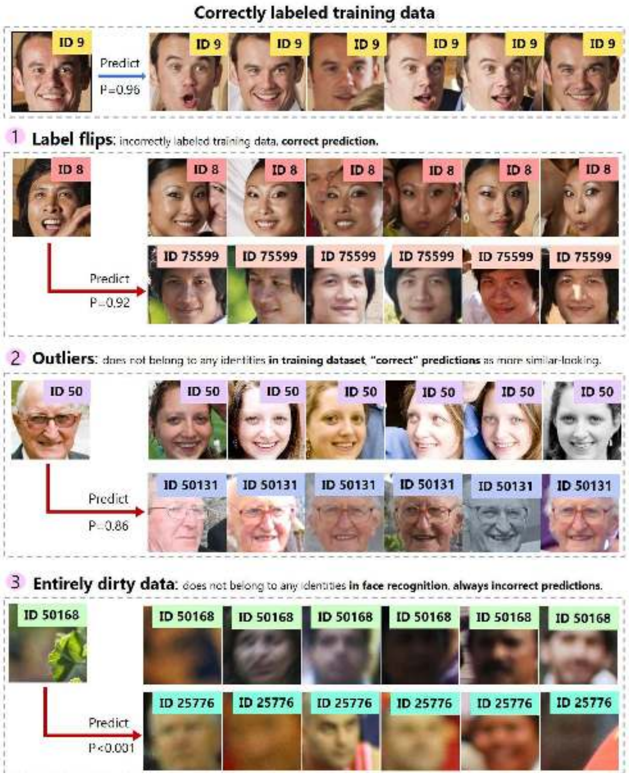

Figure 2. Three types of noise in face training dataset. The predicted class probabilities P of training data could be used for screening the third type noise, "Entirely dirty data", because P of third type noise is still extremely small when the model is trained sufficiently. The first and second type noisy labels are likely to be eventually highly inconsistent with model predictions when the model is trained relatively sufficiently, and these predictions deserve more trust.

We use some models trained on the head data for noise analysis, which is illustrated in Figure 2. We gain two insights. First, when the model is trained relatively well, the predicted class probabilities P of training data could be used for screening the third type noise, "Entirely dirty data". Because P of third type noise is extremely small when the model is already trained sufficiently, while this characteristic does not occur in other types of noise or the clean training data. Second, although we can not distinguish the first and second type noise from correctly labeled training data just by the predicted probabilities P, we find another way to mitigate them. Inspired by [27], we find that in the training process, model predictions of the first and second type noise are likely to be eventually highly inconsistent with origin labels. While these model predictions deserve more trust as the base model improves over time.

All things considered, we try to mitigate the damage of three types data noise by adapting learning criteria dynamically. First, the effect of the entirely dirty training data, which we referred to as the third type noise before, is eliminated thoroughly by blocking the gradient of them. Meanwhile if the training data is not entirely dirty, we place more trust in the model prediction by incorporating a hypothetical training label (a hypothetical W in the feature space), *i.e*. the hypothetical training label is a convex combination of the origin label with probability ρ and the current predict class with probability $1-\rho$.

Formally, the Noise Resistance (NR) loss is defined as:

$$L_{NASB} = -\frac{1}{N}\sum\nolimits_{i=1}^{N}\Big(\alpha(P_{y_{i_p}})\log\left(P_{y_i}\right)+\beta(P_{y_i})\log\Big(P_{y_{i_p}}\Big)\Big), \tag{1}$$

where N is the number of training samples in a batch, P_{y_i} is the predict probability of "true" class and $P_{y_{i_p}}$ is that of the current predict class, $\alpha(P)$ and $\beta(P)$ control the degree of combination:

$$\alpha(P)=\begin{cases}\rho, & P>t\\ 0, & P\leq t\end{cases}, \quad \beta(P)=\begin{cases}1-\rho, & P>t\\ 0, & P\leq t\end{cases}. \tag{2}$$

The hyperparameters in NR loss, ρ and t are set piecewise in the training process. That is, ρ is set to 1 and t is set to 0 at the beginning of training, and when the model is trained relatively sufficiently so that it could distinguish noise itself, ρ is reduced slightly and t is set to a small value. P_{y_i} and $P_{y_{i_p}}$ take different forms when NR loss is combined with different loss functions. In Specifically, in the Noise Resistance Softmax loss (NRS):

$$P_{y_i}=\frac{e^{W_{y_i}^T x_i+b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i+b_j}}, \quad P_{y_{i_p}}=\frac{e^{W_{y_{i_p}}^T x_i+b_{y_{i_p}}}}{\sum_{j=1}^{n} e^{W_j^T x_i+b_j}}, \tag{3}$$

where $x_i \in \mathbb{R}^d$ denotes the deep feature of the i-th samples, y_i is the "true" training label and y_{i_p} is the current predict label.

$$y_{i_p}=\arg\max_{y_j}\frac{e^{W_j^T x_i+b_j}}{\sum_{k=1}^{n} e^{W_k^T x_i+b_k}}. \tag{4}$$

The feature dimension, d is set as 512 following [7, 21, 36, 39, 44]. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weights $W \in \mathbb{R}^{d\times n}$ in the last fully connected layer, n is the identities number in the head data, and $b \in \mathbb{R}^d$ is the bias term. In Noise Resistance Large-margin Softmax loss, *e.g*. the Noise

Resistance CosFace [36] (NRC),

$$\begin{aligned} P_{y_i} &= \frac{e^{s\left(\cos\theta_{y_i} - m_C\right)}}{e^{s\left(\cos\theta_{y_i} - m_C\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}, \\ P_{y_{i_p}} &= \frac{e^{s\cos\theta_{y_{i_p}}}}{e^{s\left(\cos\theta_{y_i} - m_C\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}, \end{aligned} \tag{5}$$

and the Noise Resistance Arcface [7] (NRA),

$$\begin{aligned} P_{y_i} &= \frac{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}, \\ P_{y_{i_p}} &= \frac{e^{s\cos\theta_{y_{i_p}}}}{e^{s\left(\cos\left(\theta_{y_i} + m_A\right)\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}, \end{aligned} \tag{6}$$

where $\|x_i\|$ is re-scaled to the hypersphere radius s, m_A and m_C are the additive angular margins. We use $m_A = 0.5$, $m_C = 0.35$, and $s = 64$ following settings in [7, 36].

3.2. Joint training with the tail data

Since a relatively discriminative model have been learned on the head data, we further consider enhancing this model by exploring the complementary information in the tail. Delving into the tail data is confronted with the challenge mainly in three aspects: (1) There exits a large number of identities in the tail data. (2) Each tail identity only contains rarely samples. (3) More unfortunately, a considerable portion of them is noisy.

Considering these challenges, our primary motivation for avoiding the corresponding undesirable effects in the tail, is to extract the most credible information of the tail to enhance the face representation learned from the head data. Therefore, we devise a simple yet effective Center-dispersed Loss to deal with the tail: extract features of tail identities using the base model supervised by the powerful head data; then add the tail data gradually in an iterative way and disperse these identities in the feature space so that we can take full advantage of their modest but indispensable information.

To be more specifically, Center-dispersed (CD) Loss can be formulated as:

$$L_{CD} = \min \frac{1}{m(m-1)} \sum_{k=1}^{m(m-1)} {S_{i,j}}^2. \tag{7}$$

$S_{i,j}$ is the similarity between identity i and j in mini-batch, where the most hard m identities are mined from a candidate bag to construct a mini-batch for efficiency.

$$S_{i,j} = \left(\frac{C_i}{\|C_i\|}\right)^T \left(\frac{C_j}{\|C_j\|}\right). \tag{8}$$

The overall cost is the mean of the similarity. C_i and C_j represent identity i and j. An identity is fomulated as the center of normalized features, which can be relatively robust even to moderate noise:

Algorithm 1 The joint training process in the second stage

```
Input:
  The head data $D_h$, the tail data $D_t$, base model ($\theta_{ResNet}, W_{fc}$).
Output:
  The model ($\theta_{ResNet}, W_{fc}$).
Global parameters:
  Hyperparameter in Noisy Resistance (NR) loss $\rho$, $t$.
  Mini-batch size (NR) $s1$.
  Weight of Center-dispersed (CD) Loss $\eta$. Number of identities in a mini-batch (CD) $m$, in a candidate bag (CD) $M$. Samples of an identity $n$. So mini-batch size (CD) $s2 = m \times n$, Candidate bag size (CD) $s = M \times n$.
Initialization at the beggining of an epoch:
  // Constructing queues $Q1$ for NR and $Q2$ for CD.
  $Q1$=$Q2$={}.
  for $x_i$ in $D_h$ do
    if $P_{y_{i_p}} > t$ then
      $Q1$.append($x_i$).
    end if
  end for
  for $ID_i$ in $D_t$ do
    Randomly select $n$ samples $\{x_1, \cdots, x_n\}$ of $ID_i$.
    $Q2$.append($\{x_1, \cdots, x_n\}$).
  end for
  shuffle $Q1$ and $Q2$.
Optimization in an epoch:
  while $Q1$ is not empty do
    $B_1 \leftarrow$Take out a mini-batch with $s1$ samples in $Q1$.
    $L_{NASB}$ (1), $\nabla L_{NASB} \leftarrow B_1$.
    ($\theta_{ResNet}, W_{fc}$) $\leftarrow \nabla L_{NASB}$
  end while
  while $Q2$ is not empty do
    $C1$ $\leftarrow$Take out a candidate bag with $M$ identities in $Seq2$, extract their features using $\theta_{ResNet}$.
    $list_{ID} = \{ID_1, \cdots, ID_M\} \leftarrow$Calculate and sort $S_{i,j}^k$ (8) in $C1$.
    $B_2 \leftarrow$Constructing a mini-batch with first $m$ identities in $list_{ID}$.    // Hard Identities mining
    $L_{CD}$ (7),$\nabla L_{CD} \leftarrow B_2$.
    $\theta_{ResNet} \leftarrow \eta \nabla L_{CD}$.
  end while
```

$$C_i = \frac{1}{n} \sum_{t=1}^{n} \frac{x_t}{\|x_t\|}, n \leqslant n_i, \tag{9}$$

where x_t is feature of the t-th sample randomly selected from the tail identity i, the feature representation is learned from the head data, identity i has n_i samples and we randomly select the fix number n samples to form a mini-batch.

To avoid the feature space is damaged recklessly by the tail, the head data should always perform its responsibilities in stabilizing the model. Hence the second stage need joint training in a multi-task style, as shown in Figure 1. The CNN architectures of the two tasks are exactly the same, and the weights are shared. We summarize process of the second training stage in Algorithm 1 to precisely describe the joint training and the hard identity mining for the tail data.

4. Experiment

4.1. Experimental settings

Training Data. We evaluate our methods by performing experiments on two training dataset: (1) CASIA-WebFace [42] and its two type of long-tailed variants; (2) MegaFace Challenge 2 (MF2) [24].

Networks. Two backbone architectures are used in the following experiment. We adopt the network setting ResNet50 which was used in Arcface [7] for better convergence speed and stability. The block setting of this network is the "BN [16]-Conv-BN-PRelu-Conv-BN" structure, and the output setting is"BN-Dropout [30]-FC-BN". For fair comparison, we also use another network similar to [21], which has 64 convolutional layers and is based on residual units [11]. The models are trained with SGD algorithm, with fixed momentum 0.9 and weight decay 0.0005. In our experiment, the batch size is set to 360 for both the head data and the tail data. In the first training stage, the learning rate starts from 0.1 and is devided by 10 when the performance plateaus. While in the second stage, the learning rate is fixed to the last value in the first stage, the weight of CD loss starts from 1 and is increased gradually. The number of identities in the candidate bag is set to 600 training on variants of CASIA-WebFace and 3600 on MF2 [24] respectively. The hyperparameter ρ is set to 0.9, t is set to 0 training on variants of CASIA-WebFace and 0.007 on MF2 respectively. All the network, data iterator and the loss layer are implemented on MxNet [5].

Data Preprocessing. Following [7,21,36,39,44], we use MTCNN detecting face area and five landmarks. Then we adopt five landmarks for similarity transformation to normalize face images. After that we obtain the cropped faces which are resized to be 112 $\times$ 112. Each pixel (in [0,255]) in RGB images is normalized by subtracting 127.5 then dividing by 128. For data augmentation, horizontally flip with a probability of 50% and transformation to monochrome augmentation with a probability of 20% are used.

Testing. For testing, features of original image and the flipped image are concatenated together to compose the final face representation as [7,36] do. The similarity score is the cosine distance of features.

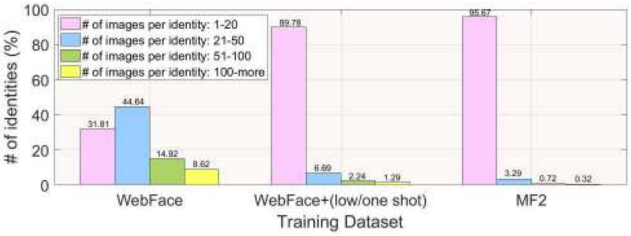

Figure 3. The distribution of imbalanced training datasets we used. "WebFace" refers to CASIA-WebFace [42], which performs actually a fusiform distribution. "WebFace+" are variants of CASIA-WebFace [42], which is composed of WebFace [42] and MS-Celeb-1M [9]. Both WebFace+ and MF2 behave long-tailed distributions.

4.2. Experiment on CASIA-WebFace and its long-tailed variants

CASIA-WebFace [42] is a dataset collected from IMDb website. The original CASIA-WebFace dataset contains 0.49M photos from 10,575 celebrities. CASIA-WebFace contains 9.3-13.0% noise according to research in [35]. Actually, CASIA-WebFace is an imbalanced database performing a fusiform distribution. The identity distribution of CASIA-WebFace is shown in Figure 3.

According to statistics, if we regard identities who have more than 10 images as the head identities, then there are 99.72% of identities have relatively sufficient images for training. To get a long-tailed training datset as testbed, besides head identities of CASIA, we add some tail identities using images from MS-Celeb-1M [9]. We experiment under two experiment settings, with low shot identities as the tail and with one shot identities as the tail. Correspondingly, two datasets are obtained, denoted as "WebFace+(low shot)" and "WebFace+(one shot)". The ratio of the head and the tail identities for both WebFace+(low shot) and WebFace+(one shot) is 10K:60K. The only difference between them is that the tail identities of WebFace+(low shot) each have 3 images while only 1 images for WebFace+(one shot). The identity distribution of WebFace+(low shot/one shot) is shown in Figure 3. Eventually, both WebFace+(low shot) and WebFace+(one shot) have 70K identities, with WebFace+(low shot) containing 0.67M images, and WebFace+(one shot) containing 0.55M images respectively. Besides, we keep the noise level in the two variants of CASIA-WebFace.

For comparison, we train models on the original CASIA-WebFace, WebFace+(low shot) and WebFace+(one shot) under supervision of softmax Loss, CosFace(LMCL) [36] and ArcFace [7]. Then we compare them with our method on the three datasets. In details, all the models are trained using the aforementioned ResNet50. The hyperparameters of softmax Loss follow SphereFace [21], while others follow the settings of original paper.

Training Data→	WebFace			WebFace+tail(one shot)			WebFace+tail(low shot)		
Method↓	LFW	CPLFW	YTF	LFW	CPLFW	YTF	LFW	CPLFW	YTF
Softmax Loss	99.25	83.72	94.74	99.37	83.32	95.10	99.15	83.10	94.94
Ours(NRS+CD)	**99.28**	**83.75**	**95.00**	**99.40**	**84.75**	**95.78**	**99.40**	**84.97**	**95.58**
CosFace(LMCL) [36]	99.55	87.67	95.52	99.47	87.85	96.12	99.50	87.25	95.60
Ours(NRC+CD)	99.43	**87.92**	**95.64**	**99.48**	**88.18**	**96.12**	99.47	**88.00**	**95.98**
ArcFace [7]	99.52	87.20	95.60	99.55	86.32	93.82	99.40	88.15	94.96
Ours(NRA+CD)	**99.55**	**87.53**	95.50	99.53	**88.03**	**96.04**	**99.55**	**88.22**	**95.76**

Table 1. Results on the controlled experiments by varying training datasets and training methods.

Method	Nets	Layer	Data	LFW	YTF
DeepFace [33]	3	6	4M	97.35	91.4
FaceNet [28]	1	14	200M	99.63	95.1
VGG Face [26]	1	16	2.6M	98.95	97.3
DeepID2+ [32]	25	-	0.3M	99.47	93.2
Baidu [20]	1	10	1.3M	99.13	-
Center Face [39]	1	7	0.7M	99.28	94.9
Noisy Softmax [4]	1	8	WebFace+	99.18	94.88
Rangeloss [44]	1	28	1.5M	99.52	93.7
Augumentation [23]	1	19	1.5M	98.06	-
Center invariant loss [41]	1	22	WebFace	99.12	93.88
Feature transfer [43]	1	-	4.8M	99.37	-
Softmax Loss	1	64	WebFace	97.88	93.1
Softmax Loss+Contrastive [31]	1	64	WebFace	98.78	93.5
Triplet Loss [28]	1	64	WebFace	98.70	93.4
L-Softmax Loss [22]	1	64	WebFace	99.10	94.0
Softmax+Center Loss [39]	1	64	WebFace	99.05	94.4
SphereFace(A-Softmax) [21]	1	64	WebFace	99.42	95.0
CosFace(LMCL) [36]	1	64	WebFace	99.33	96.1
Ours(NRC+CD)	1	50	0.55M	**99.48**	**96.12**
Ours(NRA+CD)	1	50	0.55M	**99.53**	**96.04**

Table 2. Comparison of the proposed unequal-training method with state-of-the-art method in face recognition community.

We test models on three popular face datasets, LFW [15], Cross-Pose LFW (CPLFW) [45] and YTF [40]. LFW [15] dataset contains 13233 face images from 5749 different identities. CPLFW [45] dataset is a derivative dataset of LFW, addressing cross-pose chanllenge in face recognition. YTF [40] is a database of face video collected from YouTube, which consists of 3,425 videos of 1,595 different people. We follow the unrestricted with labeled outside data protocol on all the test datasets.

Result and discussion. The results are shown in Table 1. We can see that long-tailed noisy training dataset may impair softmax, CosFace(LMCL) [36] and ArcFace [7] more or less. The reasons why they perform poorly on long-tailed noisy settings may be imbalance of decision boundaries caused by long-tailed distribution and inaccuracy of decision boundaries caused by label noise.

On the contrast, our method could still gains benefit on the long-tailed noisy training dataset mainly for three reasons: (1)The unequal-training frame offer the head data and the tail data respectively for two paralleled representation space, avoiding the imbalanced decision boundaries. (2)The hypothetical face representation of a noisy sample in NR loss (refer to Figure 1) mitigate the negative effects of noisy samples to optimization of deep model. (3)The center of normalized features in proposed CD loss, which is the most credible information in the tail data, is relatively robust to noise. Furthermore, except for FaceNet [28] trained on 200M data, our method is shown to outperform all the previous methods listed in Table 2.

4.3. Experiment on MegaFace Challenge 2 (MF2)

Our final target is to obtain the state-of-art performance on a real world long-tailed noisy dataset, MegaFace Challenge 2 (MF2): Training on 672K identities [24], which requires all algorithms to be trained on the same data with 672K identities and 4.7M photos, and tested at the million scale.

The training dataset in MF2 is from the massive collection of Creative Commons photographs released by Flickr, where most of photos are common people. MF2 is exactly an extremely unbalanced training set, where the each identity contains at most 2,469 images, at least 3 images and average 7 images per identity, 88.42% of identities have less than 10 images. The identity distribution is shown in Figure 3. Moreover, according to research in [35], MF2 contains up to 33.7-38.3% noise. This suggests that real-world collected datasets are more prone to long-tailed distribution, only limited identities apear frequently while other hundreds of thousands identities have very few samples, with significant noise inherent in the whole dataset.

Apart from the training dataset, MF2 contains gallery set and probe set. The gallery set is a subset of Flickr photos. The probe sets has two existing databases: FaceScrub [25] and FGNet [1]. According to [24], there is no overlap between the gallery set and the training set, or between the gallery set and the probe set. We use both FaceScrub and FGNet as probe sets to evaluate the performance of our method.

Evaluate our method. To find the proper percentage for the head data, by reference to the distribution of training dataset, we train models on different data where each identity have more than 8, 9, 10 images respectively. The models are trained using the aforementioned ResNet50. The results are shown in Table 3, we select MF2($\geqslant$ 9 images/id, 2.3M images and 100K identities) as the head data in the next experiment, denoted as MF2-h9. Correspondingly,

Training data	FaceScrub Rank1 acc.	FGNet Rank1 acc.	LFW
MF2($\geqslant$ 10 images/id, 2.1M images and 86 K identities)	76.24	55.51	99.57
MF2($\geqslant$ 9 images/id, 2.3M images and 100K identities)	**78.97**	**58.26**	**99.58**
MF2($\geqslant$ 8 images/id, 2.4M images and 121K identities)	78.32	57.14	99.57

Table 3. We select MF2($\geqslant$ 9 images/id, 2.3M images and 100K identities) as the head data, denoted as MF2-h9. The rest of the data is the tail data, denoted as MF2-t9. "Rank1 acc." refers to the rank-1 face identification accuracy under 1M distractors on MF2.

Training data and Method	FaceScrub Rank1 acc.	FaceScrub ver.	FGNet Rank1 acc.	FGNet ver.	LFW
Softmax(Rolling, MF2-h9+MF2-t9)	42.87	52.75	17.96	10.21	98.03
Softmax(MF2-h9)	54.72	66.41	27.71	25.01	98.85
NRS(MF2-h9)	55.38	65.33	26.90	20.14	98.95
NRS(MF2-h9)+CD(MF2-t9)	**57.50**	**66.08**	**31.96**	**31.56**	**99.05**
Arcface(Rolling, MF2-h9+MF2-t9)	75.25	85.29	52.67	54.67	99.28
Arcface(MF2-h9)	78.97	88.07	58.26	59.24	99.58
NRA(MF2-h9)	79.52	88.55	59.89	60.09	99.45
NRA(MF2-h9)+CD(MF2-t9)	**80.02**	**89.93**	**60.39**	**60.99**	**99.52**

Table 4. Face identification and verification evaluation on MF2(ResNet50). "Rank1 acc." refers to the rank-1 face identification accuracy under 1M distractors and "ver." refers to face verification TAR(True Accepted Rate) at 10^{-6} FAR(False Accepted Rate).

Method	Protocol	Rank1 Acc.	Ver.
SphereFace(A-Softmax) [21]	Large	71.17	84.22
CosFace(LMCL) [36]	Large	74.11	86.77
Rangeloss [44]	Large	69.54	82.67
LMLE [13]	Large	74.76	87.78
CLMLE [14]	Large	76.26	89.41
Ours(ResNet64)	Large	**78.12**	88.03

Table 5. Comparison with 64-layer ResNet results. "Rank1 acc." refers to the rank-1 face identification accuracy under 1M distractors and "ver." refers to face verification TAR(True Accepted Rate) at 10^{-6} FAR(False Accepted Rate).

Method	Protocol	Rank1 Acc.	Ver.
3DiVi	Large	57.05	66.46
NEC	Large	62.12	66.85
GRCCV	Large	75.77	74.84
Yang Sun	Large	75.79	84.03
CosFace(LMCL) [36]	Large	74.11	86.77
Ours(ResNet50)	Large	**80.02**	**89.93**

Table 6. Comparison with top results on MF2 Leaderboard. "Rank1 acc." refers to the rank-1 face identification accuracy under 1M distractors and "ver." refers to face verification TAR(True Accepted Rate) at 10^{-6} FAR(False Accepted Rate).

identities containing less than 9 images in MF2 are specified as the tail data, which is denoted as MF2-t9. We perform our unequal-training framework supervised by NRS(A) and CD loss, by finetuning from base models trained on MF2-h9.

For comparison, we train rotating softmax and Arcface models on all the training data of MF2, which is split equally into 8 subset with about 80K identities. This training methods are inspired by Model C in [24], but we enhance these methods by training all the data equally and adding the training epoch. The result of models trained using ResNet50 are shown in Table 4. Models trained on the head data outperform than those trained on all the long-tailed data, which is consistant with empirical investigation in [44]. While our method delve more correct information in the long-tailed dataset, achieving extra improvement over models training on both classical softmax and large margin softmax.

Comparison with existing methods. For comparison with existing state-of-art methods tackling long-tailed problem, we train MF2 using the ResNet64 achitecture similar to [21]. We adopt Batch Normalization [16] because it is hard to guarantee the convergence stability using strong constraint loss in large scale dataset. The result is shown in Table 5, besides, some competitive results on MF2 Leaderboard are also listed in Table 6. The result demonstrates the superiority of our method in training on real-world long-tailed noisy face dataset. In particular, our method obtains the top performance on MegaFace Challenge 2 (MF2): Training on 672K identities [24].

5. Conclusion

In this paper, to address training on long-tailed noisy dataset, we propose an unequal-training framework and new supervision loss functions, Noise Resistance (NR) loss and Center-dispersed (CD) loss. By dealing with the head data and the tail data respectively according to the distribution, we take full advantage of their respective characteristics. Our method achieves the new state-of-the-art performance on existing face benchmarks.

6. Acknowledgments

This work was supported by Canon Information Technology (Beijing) Co., Ltd. under Grant No. OLA18001.

References

[1] Fg-net aging database. http://www.fgnet.rsunit.com/.

[2] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2):31:1–31:50, Aug. 2016.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.

[4] Binghui Chen, Weihong Deng, and Junping Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, 2017.

[5] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015.

[6] Yuheng Chen, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[7] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv:1801.07698*, 2018.

[8] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. Marginal loss for deep face recognition. In *CVPR Workshops*, 2017.

[9] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[10] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220 – 239, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv:1709.01507*, 2017.

[13] Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

[14] Chen Huang, Yining Li, Change Loy Chen, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *arXiv:1806.00194*, 2018.

[15] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

[17] Ming Ting Kai. A comparative study of cost-sensitive boosting algorithms. In *ICML*, 2000.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *arXiv:1804.08348*, 2018.

[20] Jingtuo Liu, Yafeng Deng, Tao Bai, and Chang Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015.

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[22] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[23] Iacopo Masi, Anh Tuan Tran, Jatuporn Toy Leksut, Tal Hassner, and Gérard G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *ECCV*, 2016.

[24] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *CVPR*, 2017.

[25] Hong Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2015.

[26] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, 2015.

[27] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv:1412.6596*, 2014.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[31] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[32] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[33] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[34] Chong Wang, Xipeng Lan, and Xue Zhang. How to train triplet networks with 100k identities? In *ICCV Workshops*, 2017.

[35] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. *arXiv:1807.11649*, 2018.

[36] 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 *CVPR*, 2018.

[37] Mei Wang and Weihong Deng. Deep face recognition: A survey. *arXiv:1804.06655*, 2018.

[38] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NIPS*. 2017.

[39] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. *A Discriminative Feature Learning Approach for Deep Face Recognition*. 2016.

[40] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

[41] Yue Wu, Hongfu Liu, Jun Li, and Yun Fu. Deep face recognition with center invariant loss. In *ACM Multimedia*, 2017.

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

[43] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for deep face recognition with long-tail data. In *arXiv:1803.09014*, 2018.

[44] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *CVPR*, 2017.

[45] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018.