Neurocomputing 330 (2019) 48–55

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A feature learning approach for face recognition with robustness to noisy label based on top-*N* prediction

Menglong Yang[a], Feihu Huang[b], Xuebin Lv[b,*]

[a] School of Aeronautics and Astronautics, Sichuan University, Chengdu 610064, PR China
[b] School of Computer Science and Engineering, Sichuan University, Chengdu 610064, PR China

ARTICLE INFO

Article history:
Received 5 May 2018
Revised 13 October 2018
Accepted 15 October 2018
Available online 16 November 2018

Communicated by Dr. Ran He

Keywords:
Face recognition
Deep learning
Noisy label

ABSTRACT

Collecting a vast amount of face data with identity labels to train a convolutional neural network is an effective mean to learn a discriminative feature representation for face recognition. However, the datasets with larger scale often contain more noisy labels, that directly affects the ultimate performance of the learned model. This paper proposes an end-to-end feature learning method with robustness to noisy label. First, a data filtering method is proposed to automatically online filter out the data with false label, by checking the consistency between the annotated label and the results of top-*N* prediction. Then the loss functions of softmax and center loss are simply revised to only supervise the reserved feature. Finally, we use MS-Celeb-1M dataset, which contains massive noisy labels, to train a 128-D feature representation without any pre-train or data pre-clean. A single learned model gets an accuracy of 99.43% on LFW test set, that is very close to the model trained using the clean data.

1. Introduction

Deep learning, especially convolutional neural networks (CNNs), have achieved great success in many fields of computer vision, such as salient object detection [1,2], object classification [3], object detection [4–6], object tracking [7–9], object classification [10,11], stereo matching [12,13], photo cropping [14] and so on. Face recognition, as a typical problem in computer vision, has naturally made a great success. As technologies of deep learning progress, many CNN-based face recognition algorithms [15–18] achieve the performances beyond the ability of humans, benefiting from the development of the parallel computing hardware and the large-scale training data.

In a typical face recognition algorithm, a source face image is often mapped into a feature vector by learning the complex data distribution from the training data. A discriminative feature representation is important for a face recognition system, especially an open-set identification system. Based on the fundamental criterion of shortening intra-class distance and enlarging inter-class distance, many feature learning algorithms [19–21] use some supervised loss functions to train a discriminative face representation and achieve the state-of-the-art performances.

To learn the more complex distribution, the CNNs are deeper and deeper, and more and more training data are required. Several face datasets, such as CASIA-WebFace [22] and MS-Celeb-1M [23], are automatically or semi-automatically constructed by collecting the face images from Internet or movies. Unfortunately, the large-scale datasets without careful manual reviews may contain massive noisy labels, as shown as Fig. 1, that directly affects the ultimate performance of the learned model, as shown in Fig. 2(a). But the issue of noisy label is seldom considered in the state-of-the-art face recognition algorithms.

This paper presents an feature learning approach training on the large-scale data with massive noisy labels, as Fig. 2(c) shows. The proposed method dynamically filters out the “dirty” data during the training. Without data pre-clean and secondary training, the face features learned from the “dirty” data with massive noisy label could be separable and discriminative, using the proposed method in an end-to-end way. Specifically, we use MS-Celeb-1M [23] to train the CNNs, and achieve comparable accuracy with the models trained using “clean” data. The major contributions of this paper are summarized as follows.

(1) We proposed a feature learning method with robustness to noisy label, where a data filtering algorithm is used to automatically online filter out the data with false label, by online predicting top-*N* classes and checking the consistency between the annotated label and predicted top-*N* labels.

* Corresponding author.
E-mail address: lvxb@scu.edu.cn (X. Lv).

https://doi.org/10.1016/j.neucom.2018.10.075
0925-2312/

Fig. 1. A selected example of MS-Celeb-1M dataset. All of the images are annotated as a same label "m.0b_1sx". The faces are detected by an SSD-based face detector.

(2) We revised the loss functions of softmax and centerloss that only supervise the reserved features. The features with correct label are directly sent forward to softmax loss layer and center loss layer without secondary forward calculation of the network.
(3) We train a deep face representation using MS-Celeb-1M dataset without any pre-train or data pre-clean. A single learned model with an 128-D feature representation gets an accuracy of 99.43% on LFW test set [24], without the preprocess of landmark-based alignment, which achieves a performance close to the model trained using the clean data.

2. Related Work

2.1. Feature learning in face recognition

Face recognition has achieved a series of breakthroughs as the development of deep learning. Recent years, some architectures of CNNs are proposed to learn the complex distribution, such as Inception [25], ResNet [10], DenseNet [11], attention-aware deep structure [26] and their variant versions. Aiming at learning a discriminative feature representation, the state-of-the-art algorithms learn a model to shorten intra-class distance and enlarge inter-class distance by using a loss function to supervise the CNNs. For example, Chopra et al. used contrastive loss to supervise the siamese networks [27], in order to drive the dissimilarity metric to be small for positive pairs, and large for the negative pairs. A similar strategy was used in [28], where the distance of each positive face pair is expected to be less than a smaller threshold and the distance of each negative pair is expected to be higher than a larger threshold. Schroff et al. [19] used 200 million face images

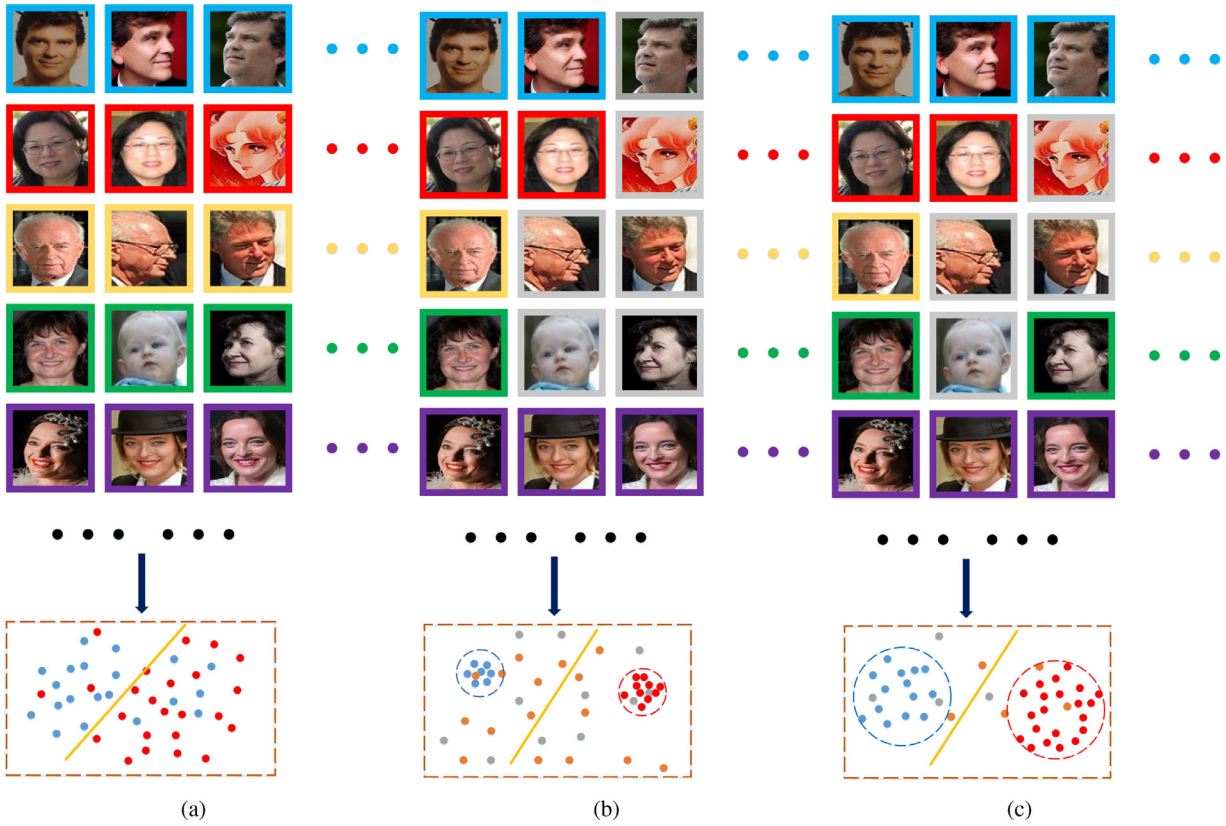


Fig. 2. (a) Noisy label makes the training difficult to converge, and the learned features are unseparable. The feature samples of the two class are respectively shown in blue and red. (b) The data filtering method with top-1 prediction. The data with false label are correctly filtered out, but an amount of data with correct label are wrongly filtered out. The faces which are filtered out are put in gray frames, and the filtered features are respectively shown in orange and gray. (c) The data filtering method with top-N prediction (an example of $N = 200$). The data with false label are correctly filtered out, and most of data with correct label are reserved.

and a triplet loss to train a unified face embedding, where maximal intra-class distance is expected to be smaller than minimal inter-class distance, and achieved the state-of-the-art performance on LFW. To further improve the separability of face feature, some algorithms, e.g., [15,29], use CNNs supervised by softmax loss function. More and more algorithms use multiple loss functions to supervise the CNNs. For example, the joint supervision of contrastive loss and softmax loss was used to train a CNN [30].

Contrastive loss and triplet loss suffer from data expansion when constituting the pairs/triplets from the training set, although they are beneficial to improve the discrimination of the face feature. To escape the problem of data expansion, Wen et al. proposed center loss [20] and obtained promising results comparable with contrastive/triplet loss. Liu et al. [21] used the proposed angular softmax (A-softmax) loss to learn angularly discriminative features, also without constituting the pairs/triplets.

2.2. Noisy label problem

In [31], Frénay et al. classify the methods to take care of noisy label into three categories [32,33]. The first category [34–36] of algorithms are expected to be naturally robust to the presence of noisy label, where label noise is not really considered. The second category [37–39] of methods tries to improve the quality of training data by using filter approaches, where noisy labels are typically identified and being dealt with before training occurs. The data filter is easy to implement but relies on a learned filter, and a substantial amount of data may be removed before training. The third category [40–42] directly models label noise during learning, the advantage of which is to separate the classification model and the label noise model, that allows using information about the nature of label noise.

In deep-learning-based methods, increasing the scale of training data is an effective way to improve performance. However, the problem of noisy label may become more acute, as training sets become larger, because the annotation of large-scale data is difficult to ensure clean. Therefore, noisy label in deep learning has received attention. Mnih and Hinton [43] presented two robust loss functions for binary classification of noisy label aerial images. Reed et al. [44] extended the softmax loss function by weakly supervised training with a notion of consistency, where a convex combination of training labels and the predictions of a current model is used to dynamically update the training targets. Besides the different way of supervision, the main difference between [44] and this work is top-1 prediction is used in [44], while top-N prediction is used in this paper, which is better adapted for multi-class problem with large number of classes, as shown in Fig. 2(b) and (c). More recently, Wu et al. [17] proposed MaxFeature-Map (MFM) operation to learn a CNN for face recognition. They adopted a semantic bootstrapping method via a pretrained deep network to handle noisy labeled images in a large-scale dataset. Inconsistent labels are detected by the probabilities of top-1 predictions, and then are relabeled or removed for further training. Different with [17], this work propose an end-to-end feature learning method for noisy label problem, which does not need twice training with the strategy of pretraining and relabeling or removing the noisy data.

3. The proposed approach

The framework of our approach is shown as Fig. 3. Similar with [20], a discriminative feature representation is learned through a CNN, which is jointly supervised by softmax and center loss. However, the major difference is the different ability of handling noisy problem, i.e. the proposed approach can automatically distinguish the noisy data from the clean data, and obtain a deep face representation with much better accuracy than [20] when the training

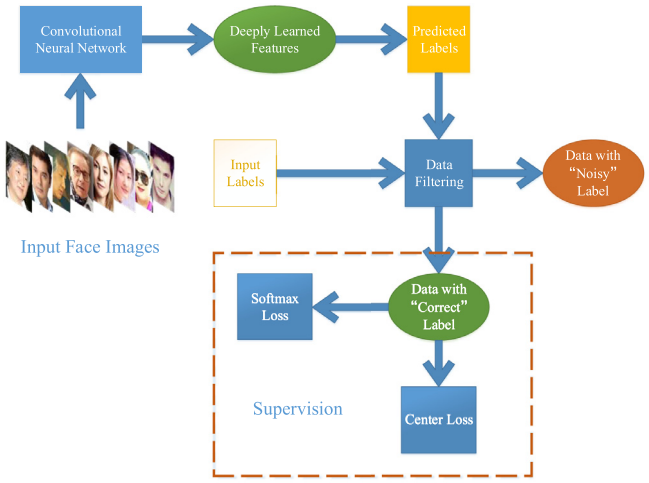


Fig. 3. The framework of the proposed deep feature learning.

set includes massive noisy labels. The feature is learned by minimizing the following loss function.

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C \tag{1}$$

where $\mathcal{L}_S$ and $\mathcal{L}_C$ denote the improved softmax and improved center loss, which are detailed described in the sections of 3.1 and 3.2, respectively. The scalar λ is a hyperparameter for balancing the two loss functions.

3.1. Softmax loss and data filtering

For learning a separable feature representation, softmax loss is widely used in many predominant CNN-based face recognition methods, and it can be written as follows.

$$\mathcal{L}_S = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} 1\{y_i = j\} \log P(\hat{y}_i = j \mid \mathbf{x}_i; W, \mathbf{b}) \tag{2}$$

where $1\{\cdot\}$ is an indicator function, so that $1\{\text{a true statement}\} = 1$ and $1\{\text{a false statement}\} = 0$, and

$$P(\hat{y}_i = j \mid \mathbf{x}_i; W, \mathbf{b}) = \frac{e^{W_j^T \mathbf{x}_i + b_j}}{\sum_{k=1}^{n} e^{W_k^T \mathbf{x}_i + b_k}} \tag{3}$$

denotes the predicted probability that the sample $\mathbf{x}_i$ belongs to jth class (i.e. the predicted label of $\mathbf{x}_i$ is $\hat{y} = j$), given the data or feature vector $\mathbf{x}_i$ and the parameters W and $\mathbf{b}$. It is abbreviated to $P(\hat{y}_i = j)$ later. Here $x_i \in \mathbb{R}^d$ (d is the feature dimension) denotes the ith deep feature with the label y_i. $W_i \in \mathbb{R}^d$ denotes the ith column of the weights $W \in \mathbb{R}^{d \times n}$ in the last fully connected layer and $\mathbf{b}$ is the bias term. The size of mini-batch and the number of class is m and n, respectively.

Under the supervision of softmax loss, the samples with noisy labels are harmful for the feature learning. Therefore, instead of calculating (2) for all the features, we expect that only the features with correct labels are substituted into softmax loss. Alternatively, we can write the softmax loss as follows.

$$\mathcal{L}_S = -\frac{1}{|X_C|}\sum_{i=1}^{m} 1\{\mathbf{x}_i \in X_C\} \log P(\hat{y}_i = j) \tag{4}$$

where X_C denotes the set of partial deep features with correct labels, and $|X_C|$ denote the number of elements of the set X_C.

The gradients of $\mathcal{L}_S$ with respect to W_j and b_j are

$$\frac{\partial \mathcal{L}_S}{\partial W_j} = -\frac{1}{|X_C|}\sum_{i=1}^{m} 1\{\mathbf{x}_i \in X_C\}\left[\mathbf{x}_i\left(1\{y_i = j\} - P(\hat{y}_i = j)\right)\right], \tag{5}$$

and

$$\frac{\partial \mathcal{L}_S}{\partial b_j} = -\frac{1}{|X_C|}\sum_{i=1}^{m} 1\{\mathbf{x}_i \in X_C\}\big(1\{y_i = j\} - P(\hat{y}_i = j)\big), \tag{6}$$

respectively.

The gradients of $\mathcal{L}_S$ with respect to $\mathbf{x}_i$ is

$$\frac{\partial \mathcal{L}_S}{\partial \mathbf{x}_i} = -\frac{1}{|X_C|}\sum_{i=1}^{m}\left[W_{y_i} - \sum_{j=1}^{n} P(\hat{y}_i = j)W_j\right], \tag{7}$$

The problem here is X_C is also an unknown. Fortunately, with the supervision of softmax and center loss, it is easy to learn a deep feature achieving a certain accuracy (verification accuracy is easy to exceed 95%), using the training set including even massive noisy labels, that is can be found in Section 4. Based on this observation, most data with noisy labels will be hopefully distinguished by some proper technique.

In this paper, we perform the feature filtering via checking the consistency between the annotated label and predicted top-N labels. A natural way is considering the label as "correct" label if the annotation is equal to the multi-class predicted label via (3), i.e., if $P(\hat{y}_i = j) > P(\hat{y}_i = k), \forall k \neq j$ and the annotated label is just $y_i = j$, then y_i is regarded as "correct". In this way, most "dirty" data can be filtered out from the deep features during the training. However, many "clean" data is also filtered out because it is very difficult to predict the label for a sample when the number of classes is very large. This phenomenon occurs common in real face recognition applications: the more the registered identities, the lower the top-1 identification accuracy. Fortunately, there is another phenomenon: the accuracy of top-N ($N \gg 1$) identification can be much higher than top-1 identification. Therefore we can utilize the top-N prediction to distinguish the "correct" label. The details can be described as follows.

Given a feature $\mathbf{x}_i$, we first compute its predicted probabilities for each class, i.e. $P(\hat{y}_i = j),\ j = 1, \ldots, n$. Then we sort these probabilities, and construct a set $\mathcal{D}$ containing the corresponding classes with top-N probabilities. Finally, if the annotated label $y_i \in \mathcal{D}$, $\mathbf{x}_i$ is regarded as "clean" feature, i.e. $\mathbf{x}_i \in X_C$, otherwise, $\mathbf{x}_i$ is considered as "dirty" feature and filtered out.

3.2. Center loss

For shortening intra-class variations, Wen et al. [20] proposed a center loss as

$$\mathcal{L}_C = \frac{1}{2m}\sum_{i=1}^{m}\|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2 \tag{8}$$

where $\mathbf{c}_i \in \mathbb{R}^d$ is the y_ith class center of deep features. The joint supervision by softmax and center loss is proved effective to make the learned deep features separable and discriminative.

For the clean data, i.e. $\mathbf{x}_i \in X_C$, it can be used to compute the center loss via (8). For the feature with wrong label, i.e. $\mathbf{x}_i \notin X_C$, it is adverse for the feature learning if substituting the wrong label into (8). We use a simple strategy handling the problem of noisy label is directly ignoring these "dirty" data, as same as the way of softmax loss (4), where the center loss is computed as

$$\mathcal{L}_C = -\frac{1}{2|X_C|}\sum_{i=1}^{m} 1\{\mathbf{x}_i \in X_C\}\|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2 \tag{9}$$

The gradients of $\mathcal{L}_C$ with respect to $\mathbf{x}_i$ is

$$\frac{\partial \mathcal{L}_C}{\partial \mathbf{x}_i} = \frac{1}{|X_C|}[1\{\mathbf{x}_i \in X_C\}(\mathbf{x}_i - \mathbf{c}_{y_i})], \tag{10}$$

The update vector of the center of each class $\mathbf{c}_j$ is

$$\Delta\mathbf{c}_j = \frac{\sum_{i=1}^{m} 1\{y_i = j\}\cdot(\mathbf{c}_j - \mathbf{x}_i)}{1 + \sum_{i=1}^{m} 1\{y_i = j\}} \tag{11}$$

To avoid large perturbations, we also use a scalar α to control the learning rate of the centers, as same as [20].

Finally we adopt the joint supervision of softmax and center loss to train the CNNs, which is given in (1). The learning details are summarized in Algorithm 1.

Algorithm 1 The proposed feature learning algorithm.

Input: Training data $\{\mathbf{x}_i\}$ and the labels $\{y_i\}$; Hyperparameters α, λ and N; The batch size m and learning rate μ^t.

Output: The parameters θ_C of CNNs; The parameters W and b in softmax loss layer; The parameters of class centers $\{\mathbf{c}_j \mid j = 1, 2, \ldots, n\}$ in improved center loss layer.

```
Randomly initialize the parameters θ_C in convolution layers and parameters W, b and {c_j | j = 1, 2, ..., n} in loss layers.
Initialize an integer N̂ = n.
while not convergent do
    t ← t + 1.
    Compute the clean data set X_C^t:
    X_C^t ← ∅.
    for i = 1 to m do
        Compute the probabilities that x_i belongs to jth class P(ŷ_i = j) for j = 1, 2, ..., n, via (3).
        Construct a label set D containing class labels with top-N̂ probabilities.
        if y_i ∈ D then
            X_C ← X_C ⋃ {x_i}.
        end if
    end for
    Compute the joint loss by L^t = L_S^t + λL_C^t, where L_S^t and L_C^t are computed according to (4) and (9), respectively.
    Compute the backpropagation error ∂L^t/∂x_i^t for each i by ∂L^t/∂x_i^t = ∂L_S^t/∂x_i^t + λ ∂L_C^t/∂x_i^t, which is referred to in (7) and (10).
    Update the parameters W by W^{t+1} = W^t − μ^t · ∂L^t/∂W^t.
    Update the parameters c_j for each j by c_j^{t+1} = c_j^t − α · Δc_j^t.
    Update the parameters θ_C by θ_C^{t+1} = θ_C^t − μ^t Σ_{i=1}^m ∂L^t/∂x_i^t ∂x_i^t/∂θ_C^t.
    Update the value of N̂ if N̂ > N.
end while
```

4. Experiment

We first introduce the implementation details, and then compare the proposed method with state-of-the-art algorithms on some publicly available testing datasets.

4.1. Implementation details

We use the open source framework Caffe [45] with our modifications to train the CNNs on a NVidia 1080 GPU with 8G built-in memory. The batch size is set to be $m = 180$, and the hyperparameters are set as $\alpha = 0.5$ and $\lambda = 0.01$ according to the recommendations in [20]. As described before, the value of N greatly affect the accuracy of top-N prediction. In the experiments, N is initially set to be equal to the number of classes n, and gradually reduced to the fixed predefined values after $50K$ iterations. We set $\hat{N} = \max\left(N, n - (0.003 * (\max(0, t - 50000)))^2\right)$, where $\hat{N}$ is a substitute of N during training, as described in the Algorithm 1. The momentum and the weight decay are set to be 0.9 and 5×10^{-4}, respetively. The learning rate is set to 0.1 initially and divided by 10

Table 1
The architectures of CNN model, where the schema of Inception-ResNet module is shown in Fig. 4. Each convolution layer is followed by a BatchNorm [46] and a ReLU layer.

Layer type	filter size/stride, pad	output size
Input layer	$160 \times 160 \times 3$	–
Conv1	3×3 / 2	$79 \times 79 \times 32$
Conv2	3×3 / 1	$77 \times 77 \times 32$
Conv3	3×3 / 1, 1	$77 \times 77 \times 64$
MaxPool1	3×3 / 2	$38 \times 38 \times 64$
Conv4	1×1 / 2	$19 \times 19 \times 80$
Conv5	3×3 / 1	$17 \times 17 \times 192$
Conv6	3×3 / 2	$8 \times 8 \times 256$
$5 \times$ Inception-ResNet modules	–	$8 \times 8 \times 256$
Conv7	3×3 / 2	$3 \times 3 \times 1792$
Average Pool1	3×3 / 1	$1 \times 1 \times 1792$
fc1	–	128

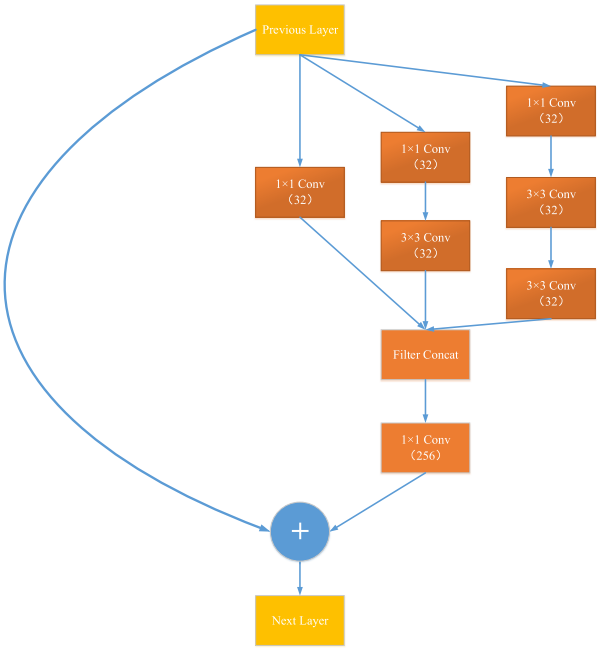


Fig. 4. The schema for 8×8 grid (Inception-ResNet module) of reduced Inception-ResNet-v1 network.

at the 150K and 300K iterations. The training is finished at 500K iterations in an end-to-end way without re-train or fine-tune on the testing database and roughly costs 40 hours.

4.1.1. The CNN architecture

To evaluate the effectiveness of the proposed method, we use unified CNN architecture in all the experiments, i.e. a reduced variant of Inception-ResNet-v1 as shown in Table 1, which is smaller and shallower than the original version in [25]. There are five same Inception-ResNet blocks as shown in Fig. 4. The output of the fully connect layer "fc1" is final 128-D face feature, that is used to calculate the similarity between the face images.

The convolution neural networks architecture is important in deep learning, and better architecture may bring better performance. This paper focus on the capability of handling the noisy data and we compare the performances of the proposed method with some state-of-the-art methods. We also need to compare the performances with different hyper-parameters. We thus use a reduced version of Inception-ResNet-v1 as the body architecture for its fast inference speed.

4.1.2. Preprocessing

Without any landmark-based alignment, an extremely simple preprocessing is adopted in the experiments. We only use an SSD-based face detector to detect the face images, and then crop the detected faces to 160×160 RGB images, which are directly sent to training or testing after normalization operation, as Fig. 1 shows. The normalization contains only subtracting a RGB value of {104, 117, 123} and dividing by 128 for each pixel of the input images.

4.1.3. Training data

We use MS-Celeb-1M [23] as the training set, which contains about 10*M* web images of 100*K* celebrities. After removing the images with identities appearing in LFW test set and detecting the faces from the images, it roughly has 8*M* face images of 95,016 unique persons.

For verifying the robustness for noisy label, we also use a clean training data, which is referred to later as clean data, where a clean list for MS-Celeb-1M provided by [17] is used to filter and relabel the face images. After removing the images overlapped with LFW test set and processed by the face detector, the clean data contains about 5*M* face images of 75,467 unique persons.

4.1.4. Test settings

We use single model for all the testing. The deep features are computed via learned model, and the score is calculated by the Euclidean distance of two features without any further processing, such as PCA, joint Bayes etc. A simple threshold comparison is finally adopted for face verification.

4.2. Effectiveness of data filtering

We first verify the effectiveness of the proposed data filtering method through comparing the accuracies on LFW of the models with and without the proposed data filtering module. Then we test the performances of our model with different values of parameter *N*. In the experiments, we use the same CNN architecture, same preprocessing method and same training data (MS-Celeb-1M) for fair comparison.

We set the parameter $N = 200$ and study the effectiveness of the data filtering method, as shown in Table 2. Due to massive noisy labels, the accuracy testing on LFW is 96.15% if using only traditional softmax loss to supervise the CNN. But if we use the proposed data filtering method, the accuracy is improved to 97.42%. Similarly, if using the joint supervision of softmax and center loss, the accuracy is improved from 96.78% (without data filtering) to 99.43% (with data filtering).

Then we use joint supervision of softmax and center loss to train the CNNs, and evaluate the performances of the learned models with different values of *N*, and the results are shown in Table 3. From the results comparison, we can see that the proposed method has certain tolerance for the value of *N*. But if *N* is too large or too small, such as $N = 1$ or $N = 5000$, the learned models performs significantly lower. In fact, if *N* is too large, the probability of data with false label is wrongly reserved is large, that means the parameters of the CNN model are learned from too many "dirty" data. Conversely, if *N* is too small, the probability of data with correct label is wrongly filtered out is large, that means the scale of training samples is greatly reduced.

5. Discussion

There are four cases for data filtering, i.e. (1) clean data is reserved, (2) dirty data is filtered out, (3) clean data is filtered out, and (4) dirty data is reserved. Cases (1) and (2) are correct and expected, (3) and (4) are wrong and undesired. Suppose the accuracy of top-*N* prediction is *r*, the rate of noisy label is η, and

Table 2
Effectiveness evaluation of the proposed data filtering method on LFW dataset.

Method	N	Accuracy
Softmax	–	96.15%
Softmax	200	97.42%
CenterLoss	–	96.78%
Our model	200	99.43%

Table 3
Comparison the models with different values of N on LFW dataset.

N	1	10	50
Accuracy	98.8%	99.15%	99.23%
N	100	200	500
Accuracy	99.37%	**99.43%**	99.4%
N	1000	2000	5000
Accuracy	99.3%	99.27%	98.95%

the predicted multi-class rank is totally random for all the "dirty" samples, then the rate of data are approximately (1) $(1-\eta)\cdot r$, (2) $\eta\cdot(1-\frac{N}{n})$, (3) $(1-\eta)\cdot(1-r)$ and (4) $\eta\cdot\frac{N}{n}$ for the four cases, respectively.

Let us simply analyze the error probability of filtering the data. Given a sample with correct label, the probability that it is mistakenly filtered out is approximately $1-r$ (case 3), and given a sample with wrong label, the probability that it is mistakenly reserved is approximately $\frac{N}{n}$ (case 4).

For example, we have 100,000 identities, i.e. $n = 100,000$. We first consider the case of $N = 1000$. The accuracy of top-1000 prediction is assumed to be $r = 0.9$, and half of the training data is falsely labeled, i.e. $\eta = 0.5$. Then only about 10% of samples with correct label are filtered out, and only about 1% of samples with wrong label are reserved, which is tolerable for the face feature learning. In contrast, if we do not perform the data filtering, there are about 50% of samples with wrong label reserved.

Then we consider the case of $N = 1$. When $n \gg N$, $\frac{N}{n}$ is almost equal to 0, and the probability that a sample with wrong label is mistakenly reserved is negligibly small. However, a sample with correct label is probably mistakenly filtered out, because the accuracy top-1 prediction may be very low when the number of classes is large. In general, larger value of N results in higher accuracy of top-N prediction. Intuitively, as the learner improves over time, the accuracy of top-N prediction is higher and the operation of data filtering can be trusted more. Therefore, we adopted a training strategy that N is initially set to be large and gradually reduced to a fixed value.

5.1. Results on LFW

In this section, we compare the proposed method with state-of-the-art algorithms on the commonly used LFW [24] dataset, which contains 13,233 web-collected images from 5749 different identities, with large variations in pose, expression and illuminations. We test on 6000 face pairs following the standard protocol of unrestricted with labeled outside data.

The results comparison is shown in Table 4. In Table 4, CenterLoss-v2 and CenterLoss-v3 are the models we trained on MS-Celeb-1M using the same CNN architecture and preprocessing with the proposed method, so that we can fairly compare the proposed method with them. From the results comparison, we have the observation that the proposed method has good robustness for noisy label. The learned models achieve the accuracy of 99.43%, which is comparable with the models learned on clean data, i.e. 99.45% of CenterLoss-v2. In order to further show the effectiveness of the proposed method, we replace the body architecture of CNN with DenseNet-121 [11], and train the network with $N = 200$. We get a accuracy of 99.52%, close to the performance of the model learned with clean data, i.e. 99.55%. The corresponding methods are named as "DenseNet" and "Our model with DenseNet" in Table 4, respectively. Although the DenseNet-121 [11] performs better, but the inference speed is about four times slower than the reduced variant of Inception-ResNet-v1 [25] in our experiments. Therefore, only the reduced variant of Inception-ResNet-v1 [25] is evaluated in other experiments.

Note we do not use any landmark-based alignment and fine-tune the model on LFW, and the training data is much less than several private business methods, such as DeepFace [29] and FaceNet [19]. Light CNN-29 [17] is also leaned on dirty data and performs well, but it uses semantic boostrapping method to repeatedly relabel the training dataset and re-train the model, while this paper use an end-to-end training method.

5.2. Results on MegaFace

We also test our learned model on a very challenging benchmark, MegaFace database [47], which aims at the evaluation of face recognition algorithms at million-scale and consists of more than 1 million images from 690K different individuals, as a subset of Flickr photos [48] from Yahoo. MegaFace includes gallery set and probe set. The probe set contains two existing databases, i.e. Facescrub [49] and FGNet [50]. Facescrub dataset contains 100K photos of 530 unique individuals (55,742 images of males and 52,076 images of females). FGNet dataset is a face aging dataset with 1002 images from 82 identities. Each identity has multiple face images at different ages (ranging from 0 to 69). In this experiment, we only tests on one of the three gallery set (set 1) for both face identification and face verification protocols, with the provided

Table 4
Comparison with the state-of-the-art methods on LFW dataset. The unrestricted protocol follows the LFW unrestricted setting and the unsupervised protocol means the model is not trained or fine-tuned on LFW in supervised way. The preprocess of "alignment" means the method uses landmark-based alignment, while "crop" means the model uses only the simple cropping operation.

Method	Outside data	Purity of data	Preprocess	#Nets	protocol	Accuracy
DeepFace [29]	4.4M (private)	clean	alignment	7	unrestricted	97.35%
DeepID2+ [18]	300K (private)	clean	alignment	25	unrestricted	99.47%
FaceNet [19]	200M (private)	clean	crop	1	unrestricted	**99.63%**
CenterLoss [20]	WebFace	clean	alignment	1	unsupervised	98.70%
CenterLoss-v2	MS-Celeb-1M	clean	crop	1	unsupervised	99.45%
DenseNet[11]	MS-Celeb-1M	clean	crop	1	unsupervised	99.55%
CenterLoss-v3	MS-Celeb-1M	dirty	crop	1	unsupervised	96.78%
Light CNN-29 [17]	WebFace + MS-Celeb-1M	dirty	alignment	1	unsupervised	99.33%
Our model	MS-Celeb-1M	dirty	crop	1	unsupervised	**99.43%**
DenseNet [11]	MS-Celeb-1M	dirty	crop	1	unsupervised	97.18%
Our model with DenseNet	MS-Celeb-1M	dirty	crop	1	unsupervised	**99.52%**

Table 5
Comparison with the state-of-the-art methods on MegaFace dataset.

Method	Rank-1	VR@FAR=10^{-6}
NTechLAB	73.3%	85.081%
3DiVi Company	33.705%	36.927%
FaceNet v8 [19]	70.496%	86.473%
CenterLoss [20]	65.234%	76.510%
CenterLoss-v2	72.664%	82.916%
Light CNN-29 [17]	73.494%	84.731%
Our model	72.102%	82.280%

code[1]. For the model learned from the dirty data, we only evaluate the model with $N = 200$, and the proposed method achieve a comparable result with state-of-the-art methods, as shown in Table 5.

6. Conclusions

Aiming at the problem of noisy label in face recognition, this paper presented a simple but effective method of learning discriminative face feature from large-scale "dirty" training data. The experimental results showed the robustness of the proposed method against noisy label. With a single CNN model learned from MS-Celeb-1M, containing massive noisy labels, it can achieve the accuracy of 99.43% on LFW, which is very close to the model learned from the clean data, i.e. 99.45%. In addition, it is easy to implement. It only requires some minor modifications on softmax and center loss.

However, the biggest shortcoming of the proposed method is that the parameter N need to preset in the top-N prediction. Although there is a certain degree of tolerance for the value of N, but it is more or less inconvenienced. The second disadvantage is the dirty data make almost no contribution to the feature learning. How to automatically set an appropriate value for N and how to take full advantage of dirty data is our next work.

Acknowledgments

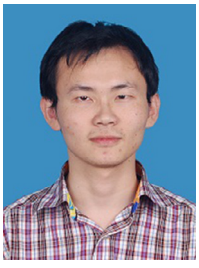
This work is supported by the funding from Sichuan Science and Technology Program (Grant No. 18YYJC1287), the funding from Sichuan University (Grant No. 2018SCUH0042), National Key Research and Development Program of China (Grant No. 2016YFC0801100), National Major Instrument Special Fund (Grant No. 2013YQ490879) and National Natural Science Foundation of China (Grant No. 61402307).

References

[1] W. Wang, J. Shen, Deep visual attention prediction, IEEE Trans. Image Process. 27 (5) (2018) 2368–2378.
[2] W. Wang, J. Shen, X. Dong, A. Borji, Salient object detection driven by fixation prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1711–1720.
[3] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, neural information processing systems (2012) 1097–1105.
[4] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[5] J. Redmon, S.K. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, Comput. Vis. Pattern Recognit. (2016) 779–788.
[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 21–37.
[7] N. Wang, D. Yeung, Learning a deep compact image representation for visual tracking, Neural Inf. Process. Syst. (2013) 809–817.
[8] D. Held, S. Thrun, S. Savarese, Learning to track at 100 fps with deep regression networks, in: Proceedings of the European Conference Computer Vision (ECCV), 2016.
[9] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking, in: Proceedings of the European Conference Computer Vision (ECCV), 2016, pp. 850–865.
[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[11] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[12] J. Zbontar, Y. LeCun, Computing the stereo matching cost with a convolutional neural network, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[13] M. Yang, Y. Liu, Z. You, The euclidean embedding learning based on convolutional neural network for stereo matching, Neurocomputing 267 (6) (2017) 195–200.
[14] W. Wang, J. Shen, Deep cropping via attention box prediction and aesthetics assessment, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.
[15] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1891–1898.
[16] C. Ding, D. Tao, Robust face recognition via multimodal deep face representation, IEEE Trans. Multimed. 17 (11) (2015) 2049–2058.
[17] X. Wu, R. He, Z. Sun, A lightened CNN for deep face representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[18] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2892–2900.
[19] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.
[20] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 499–515.
[21] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: Deep hypersphere embedding for face recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
[22] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, Clin. Orthopaed. Relat. Res. (2014). abs/1411.7923
[23] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: a dataset and benchmark for large-scale face recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 87–102.
[24] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Technical Report 07–49, University of Massachusetts, Amherst (2007).
[25] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: National Conference on Artificial Intelligence, 2016, pp. 4278–4284.
[26] W. Wang, J. Shen, L. Shao, Video salient object detection via fully convolutional networks, IEEE Trans. Image Process. 27 (1) (2018) 38–49.
[27] S. Chopra, R. Hadsell, Y. Lecun, Learning a similarity metric discriminatively, with application to face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 539–546.
[28] J. Hu, J. Lu, Y.P. Tan, Discriminative deep metric learning for face verification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1875–1882.
[29] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[30] Y. Sun, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, Adv. Neural Inf. Process. Syst. 27 (2014) 1988–1996.
[31] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[32] M. Pechenizkiy, A. Tsymbal, S. Puuronen, O. Pechenizkiy, Class noise and supervised learning in medical domains: The effect of feature extraction, in: Proceedings of the IEEE Symposium on Computer-Based Medical Systems, 2006, pp. 708–713.
[33] N. Manwani, P.S. Sastry, Noise tolerance under risk minimization, IEEE Trans. Cybern. 43 (3) (2013) 1146.
[34] C.M. Teng, Dealing with Data Corruption in Remote Sensing, Springer Berlin Heidelberg, 2005.
[35] P.L. Bartlett, M.I. Jordan, J.D. Mcauliffe, Convexity, classification, and risk bounds, J. Am. Stat. Assoc. 101 (473) (2006) 138–156.
[36] E. Beigman, B.B. Klebanov, Learning with annotation noise, in: ACL 2009, Proceedings of the Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing of the Afnlp, 2–7 August 2009, Singapore, 2009, pp. 280–287.
[37] J. Sun, F. Zhao, C. Wang, S. Chen, Identifying and correcting mislabeled training instances, in: Future Generation Communication and Networking, 2007, pp. 244–250.
[38] J. Thongkam, G. Xu, Y. Zhang, F. Huang, Support vector machine for outlier detection in breast cancer survivability prediction, in: Advanced Web and Network Technologies, and Applications, 2008, pp. 99–109.
[39] P. Jeatrakul, K.W. Wong, C.C. Fung, Data cleaning for classification using misclassification analysis, J. Adv. Comput. Intell. Intell. Inf. 14 (3) (2010) 297–302.

[1] http://megaface.cs.washington.edu/participate/challenge.html.

[40] N.D. Lawrence, Estimating a kernel fisher discriminant in the presence of label noise, in: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 306–313.
[41] C.D. Paulino, P. Soares, J. Neuhaus, Binomial regression with misclassification., Biometrics 59 (3) (2003) 670–675.
[42] T. Swartz, Y. Haitovsky, A. Vexler, T. Yang, Bayesian identifiability and misclassification in multinomial data, Can. J. Stat. 32 (3) (2004) 285–302.
[43] V. Mnih, G. Hinton, Learning to label aerial images from noisy data, in: Proceedings of the International Conference on Machine Learning, 2012.
[44] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, A. Rabinovich, Training deep neural networks on noisy labels with bootstrapping, in: Proceedings of the ICLR Workshop, 2015.
[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.
[46] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.
[47] I. Kemelmachershlizerman, S.M. Seitz, D. Miller, E. Brossard, The megaface benchmark: 1 million faces for recognition at scale, in: Proceedings of the Computer Vision and Pattern Recognition, 2016, pp. 4873–4882.
[48] B. Thomee, D.A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L.J. Li, The new data and new challenges in multimedia research, Commun ACM 59 (2) (2015) 64–73.
[49] H.W. Ng, S. Winkler, A data-driven approach to cleaning large face datasets, in: Proceedings of the IEEE International Conference on Image Processing, 2015, pp. 343–347.
[50] Fg-net aging database, (http://www.fgnet.rsunit.com/). 2010.



Menglong Yang received the B.S. degree in the School of Chemical Engineering and M.S. degree in the School of Computer Science and Engineering from Sichuan University in 2005 and 2008, respectively. He is currently an associate professor of the School of Aeronautics and Astronautics, Sichuan University. He has published more than 20 journal papers. Now he is an engineer in Wisesoft Co. His research interests include computer vision, pattern recognition and machine learning.

Feihu Huang received the B.S. degree in the College of Software Engineering and M.S. degree in the College of Computer Science and Engineering from Sichuan University in 2007 and 2014, respectively. He is currently a Ph.D. candidate at the National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University.His research direction is Computer vision and machine learning.

Xuebin Lv received the M.S. degree from Southwest Petroleum University in 2003, and the Ph.D. degree from Sichuan University in 2009, respectively. He is currently an lecturer of the School of Computer Science and Engineering, Sichuan University. Now he works in Nation Key Laboratory of Fundamental Science on Synthetic Vision of China. His research interests include data fusion, pattern recognition and machine learning.