Global-Local GCN: Large-Scale Label Noise Cleansing for Face Recognition

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

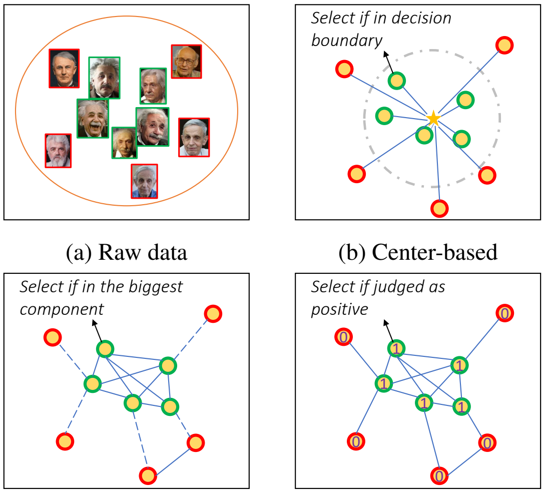
Xian Li, Dongyue Zhao, Dongchao Wen
Canon Information Technology (Beijing) Co., Ltd
{lixian, zhaodongyue, wendongchao}@canon-ib.com.cn

Abstract

In the field of face recognition, large-scale web-collected datasets are essential for learning discriminative representations, but they suffer from noisy identity labels, such as outliers and label flips. It is beneficial to automatically cleanse their label noise for improving recognition accuracy. Unfortunately, existing cleansing methods cannot accurately identify noise in the wild. To solve this problem, we propose an effective automatic label noise cleansing framework for face recognition datasets, FaceGraph. Using two cascaded graph convolutional networks, FaceGraph performs global-to-local discrimination to select useful data in a noisy environment. Extensive experiments show that cleansing widely used datasets, such as CASIA-WebFace, VGGFace2, MegaFace2, and MS-Celeb-1M, using the proposed method can improve the recognition performance of state-of-the-art representation learning methods like Arcface. Further, we cleanse massive self-collected celebrity data, namely MillionCelebs, to provide 18.8M images of 636K identities. Training with the new data, Arcface surpasses state-of-the-art performance by a notable margin to reach 95.62% TPR at 1e-5 FPR on the IJB-C benchmark.

1. Introduction

Label noise cleansing is a long-term issue in building up a dataset [9, 10, 24, 25, 30, 32]. Many studies [7, 20, 43, 44, 48, 58] point out that a noisy dataset is very harmful to model training. In recent years, the face recognition community witnesses a boost in datasets, from the first widely used deep training set CASIA-WebFace (CASIA) [54] to MS-Celeb-1M (MS1M) [21], UMDFace [8] and VGGFace2 [12]. The increasing data scale helps improve the face recognition accuracy, but the label noise also keeps increasing with their scale [43]. Therefore, to construct an effective face recognition training set, on the one hand, it is necessary to collect as many face images as possible. On the other hand, there must be a reliable data cleansing method to detect and reduce label noise in the dataset.


Select if in decision boundary
Select if in the biggest component
Select if judged as positive
0
1
1
1
1
1
0
0
0
0

(a) Raw data (b) Center-based

(c) Edge prediction (d) Node prediction (ours)

Figure 1: Three ideas for face data cleansing. (a) Face data in one identity. (b) Center-based: select images according to their distance to the identity center. (c) Graph-based edge prediction: predict edges on a graph, and pick nodes in the biggest component. (d) Graph-based node prediction: directly predict whether a node is a signal or noise on a graph.

Figure 1 compares two existing ideas with ours for cleansing large-scale face recognition datasets. As opposed to noise, the correctly labeled images are denoted as "signals". Suppose there are 10 face samples for an identity, five of which are signals (green box) and the others are noise (red box). Center-based algorithms (Figure 1b) compute the center of each identity and then select samples that are closer to the center as signals. They usually ignore the relationships between samples and rely heavily on the rate of signals [7, 54]. In fact, signals can gather together in the feature space, while noise is usually similar to only a few samples. This can be well represented by the graph structure. Considering a face image as a node on the graph, Figure 1c predicts edges between nodes according to their pairwise relationships, then picks nodes in the biggest component as signals. There is a strong assumption in this type of work [14, 36, 47] that all signals are connected. Many graph information other than node relationships is also lost. In this paper, the cleansing idea in Figure 1d is adopted, *i.e.*, directly predicts whether the nodes are signals or noise on the graph. It is more robust to big noise because of its decentralized graph structure. Further, using graph convolution techniques, the cleansing model not only learns node-wise relationships but also performs global awareness on the graph to get more discriminative node representations.

We find that a single-stage Graph Convolutional Network (GCN) can make global predictions to get good results on many graph-based tasks [13, 29, 41, 49, 53, 59], but sometimes ignores local details of the graph, causing widespread prediction errors in some difficult local regions. To solve this problem, our proposed FaceGraph leverages a cascaded framework with two stages of carefully designed GCNs, namely Global Graph Net and Local Graph Net, to make predictions in a global-to-local manner. The first network makes global graph prediction, then the second network makes local-aware refinement. To efficiently train them, a novel propagation function and some training schemes such as multi-task learning and cooperative learning are designed. In the data cleansing task, FaceGraph outperforms a single-stage GCN by a notable margin and can remove noise more accurately than previous methods.

To verify the proposed method on real data, we cleanse four widely used large-scale face datasets CASIA [54], MegaFace2 [36], MS1M [21], and VGGFace2 [12]. The effectiveness is assessed in terms of the comparative recognition performance of Arcface [15] trained on different datasets. The results show that datasets cleansed by FaceGraph effectively improve the face recognition performance compared with the ones cleansed by previous methods. Furthermore, to address the problems of low number of identities and high noise rate in the existing face datasets [43, 44], we take a great effort to collect and cleanse a large-scale face dataset, MillionCelebs, using the proposed method. The cleansed MillionCelebs dataset provides 18.8M images of 636K identities, which can largely facilitate the study of large-scale deep face recognition. For instance, the Arcface method trained by this new dataset outperforms state-of-the-art performance on the IJB-C by a notable margin.

The main contributions can be summarized as follows: (1) We propose the first GCN-based label noise cleansing method for face recognition datasets, significantly enhancing the cleansing performance. (2) A two-stage global-local GCN framework is designed with performance far beyond a single-stage network. (3) The MillionCelebs dataset is collected and cleansed to promote state-of-the-art face recognition performance and facilitate the study on large-scale deep learning. MillionCelebs is better than existing public datasets in terms of data size and the number of identities.

2. Related Work

Label Noise Cleansing. The label noise cleansing methods can be divided into graph-based and non-graph-based. Except for some manual reviewing work [8, 43], non-graph-based methods are usually straightforward and easier to manipulate, but their effects are limited. Angelova *et al.* [5] adopt data pruning to build a dataset. CASIA-WebFace [54] cleanses every subject by taking its “main photo” as a seed to accept other faces constrained by similarities and tags. VGGFace [38] and VGGFace2 [12] train SVM classifiers to reject outliers. Celeb500k [11] trains a CNN-based label predictor to select samples in a bootstrapping manner.

Differently, graph-based methods fully consider the data structure. Mode filters [18, 19] recognize noise on a graph by semi-supervised learning. RT [47] iteratively removes noise by instance pruning. MegaFace2 [28] clusters images according to the average pairwise distance in one identity. Unfortunately, most graph-based cleansing attempts have artificially designed parameters, which are hard to take full advantage of the data information. This paper develops a graph-based automatic learnable cleansing algorithm.

Graph Convolutional Networks. Following the idea of CNNs, GCN is proposed to process problems with non-Euclidean data [49]. The work on semi-supervised classification [29] provides the basic propagation formulation of a multi-layer GCN. Some work [13, 33, 41, 51, 53] modifies it to apply GCN into different categories, such as knowledge base construction and text classification. GraphSAGE [22] learns a principle of aggregation to extend GCN into inductive representation learning. GAT [42] learns a graph attention model in feature propagation. In the computer vision community, GPP [59] predicts positive neighbors in person re-identification. Zhong *et al.* [57] deploy GCN for anomaly detection. Some other work [45, 52] uses GCN to do face clustering. Different from clustering, the cleansing task needs to select one subgroup from a big group of data, while the others are dropped. In this paper, we explore the introduction of GCN into the field of face dataset cleansing.

3. Methodology

3.1. Overview

Consider a large-scale face image dataset with label noise, for instance, the celebrity images return by searching keywords on the web. Since the images are naturally

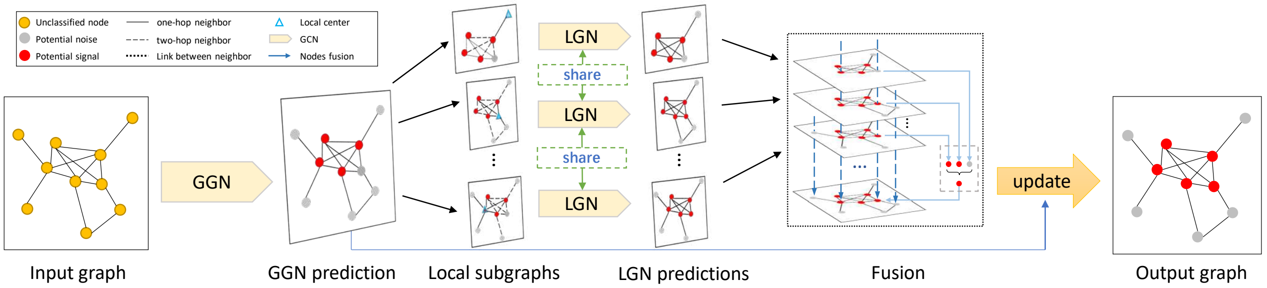


Figure 2: Overview of FaceGraph. Images in one identity are represented as nodes, and a k-NN graph is built based on deep features. A two-stage cascaded global-local cleansing is performed. In the first stage, GGN classifies all nodes globally. Based on its result, local subgraphs are built for difficult regions, and all subgraphs go through the parameters shared LGNs. The prediction results of LGNs are fused to update GGN result as the output graph. High-scoring nodes are picked as signals.

grouped by the name of the celebrity, we apply the proposed method to cleanse label noise for each group separately. This procedure largely saves manual labor to label the images. Since the performance of deep face recognition has surpassed human significantly [39], it is possible to achieve better cleansing results than manual labeling. Assume that n face samples in one identity are represented as d-dimensional $l2$-normalized features $\boldsymbol{x}_i$, $i \in \{1, 2, \cdots, n\}$. So the identity can be represented as matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n \times d}$. The cleansing task predicts labels $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n]^T$ for all n instances, where $\boldsymbol{y}_i \in \{0, 1\}$, 1 representing signals and 0 representing noise. As shown in Figure 2, FaceGraph is proposed to address the label noise problem with cascaded global-local GCNs: first making global sense (Section 3.2), then locally refining the result with rebuilt subgraphs (Section 3.3).

3.2. Global Graph Net

Global Graph Net (GGN) is a L-layer node classification graph convolutional network. Based on the pairwise cosine similarity S of feature matrix X

$$S = XX^T, \tag{1}$$

a k-NN graph $\mathcal{G}$ is built. Taking $\mathcal{G}$ as input, GGN judges nodes on the graph are signals or noise. Figure 3 shows the GGN forward propagation algorithm. The general GCN layer-wise forward propagation formula of node i is

$$\boldsymbol{h}_i^{(l+1)} = \sigma \left[F_{j \in \mathcal{N}_i} \left(\boldsymbol{h}_j^{(l)} \right) \mathbf{W}^{(l)} \right] \tag{2}$$

where $\boldsymbol{h}_j^{(l)}$ means the l-th layer output of node j, $\boldsymbol{h}_j^{(0)} = \boldsymbol{x}_j$. $\mathcal{N}_i$ is a collection of all neighbors of node i (include itself). $F : \mathbb{R}^{m \times d_{in}} \to \mathbb{R}^{d'}$ is a transforming function that transforms the features of node i and its neighbors into one feature of d' dimension, m is the number of elements in $\mathcal{N}_i$. $\mathbf{W}^{(l)} \in \mathbb{R}^{d' \times d_{out}}$ is a learnable matrix in the l-th layer. σ denotes the activation function. Therefore, the forward propagation of a node can be regarded as alternately performing the following two operations: first executing feature transformation F according to its neighbors, then passing through a fully connected layer $\mathbf{W}$ with activation σ.

Since there are big differences between the identities in a face recognition dataset, strong generalization ability is very essential to cleanse it. Following the idea of GraphSAGE [22] that learns a generalizable aggregator, we design the transforming function F in Equation 2 as

$$F_{j \in \mathcal{N}_i} \left(\boldsymbol{h}_j^{(l)} \right) = \left[\boldsymbol{h}_i^{(l)} \| Aggregate_{j \in \mathcal{N}_i} \left(\tilde{s_{ij}} \boldsymbol{h}_j^{(l)} \right) \right] \tag{3}$$

where $\tilde{s_{ij}} = \frac{S_{ij}}{\sqrt{D_i D_j}}$ is the normalized similarity score between node i and j, which appears as a weight term in the function. D_i is the degree of node i [29]. $Aggregate$ is a learnable aggregating principal function, and $\|$ is the concatenation operator. Considering that face recognition is mainly based on pairwise similarity, the similarity matrix S is used to help the aggregation process. For node i, the features of its neighbors are weighted by $\tilde{s_{ij}}$ when aggregating, so that the neighbors with low similarity to node i are forced to provide less weighted aggregating information. Then the aggregated vector is directly concatenated with $\boldsymbol{h}_i$ by a "shortcut" to obtain a $d' = 2d$ dimensional vector. This shortcut reserves the original node information along with the information from the aggregated neighbors. $Aggregate$ is designed as the sum of the outputs of a neuron:

$$Aggregate_{j \in \mathcal{N}_i} \left(\boldsymbol{h}_j^{(l)} \right) = \sum_{j \in \mathcal{N}_i} \sigma \left(\boldsymbol{h}_j^{(l)} \mathbf{A}^{(l)} + \mathbf{b}^{(l)} \right) \tag{4}$$

where $\mathbf{A}^{(l)}$ and $\mathbf{b}^{(l)}$ are deployed to learn the face aggregating principle in the l-th layer. At the last layer, we deploy $\mathbf{W}^{(L)} \in \mathbb{R}^{d' \times 1}$ and sigmoid activation to predict scores for all nodes. The nodes whose score is higher than a threshold are judged as signals. In back-propagation, Stochastic

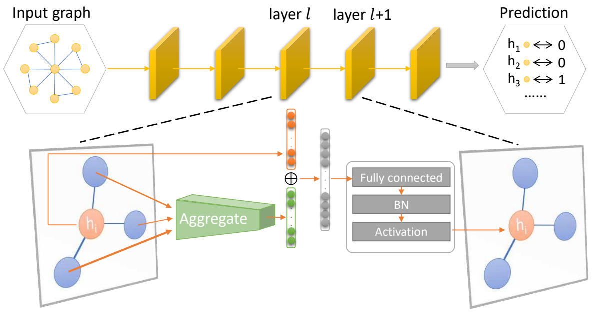


Figure 3: Global Graph Net architecture and forward propagation algorithm of node i between layer (l) and $(l+1)$.

Gradient Descent (SGD) by the binary cross-entropy loss is used. For a graph with N nodes, the GGN loss is

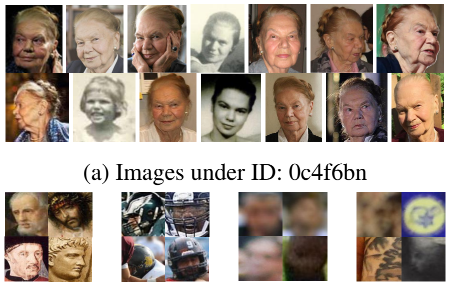
$$\mathcal{L}_G = -\frac{1}{N}\sum_{i=1}^{N}\left[\hat{\boldsymbol{y}}_i \cdot \log \boldsymbol{y}_i + (1-\hat{\boldsymbol{y}}_i) \cdot \log\left(1-\boldsymbol{y}_i\right)\right] \quad (5)$$

where $\boldsymbol{y}_i$ is the network output score of node i between 0 and 1, and $\hat{\boldsymbol{y}}_i \in \{0, 1\}$ is the label of node i.

3.3. Local Graph Net

GGN can handle most simple cases but may ignore local details on a complicated graph and output scores around 0.5 for hard nodes in some local regions, *e.g*., boundary nodes that are simultaneously adjacent to multiple signals and noise. The second stage network, Local Graph Net (LGN), is designed to solve these hard nodes. We define "low confidence nodes" as nodes that GGN outputs scores between 0.2 and 0.8, and define "high confidence nodes" the complement of low confidence nodes. Low confidence nodes are randomly selected as "local centers" on the graph, then their one-hop and two-hop neighbors are taken to build the local subgraphs. There are two special cases: (a) If one subgraph does not contain any GGN predicted signals, which means it takes very limited graph information, this subgraph is discarded. (b) If GGN predicts all nodes with high confidence, then no "centers" can be found, we pick the nodes that GGN predicts as signals to construct the only one subgraph.

All generated subgraphs go through parameters shared LGNs for subtle discrimination. LGN is designed the same as GGN in network architecture and outputs scores like GGN as well. For every node, if it is included in at least one subgraph, we obtain its final score by averaging its results from all LGNs that output scores for it. On the contrary, if it is not included in any of the local subgraphs, which means it is neither a local center nor within two-hop range of any local centers, this node is easy to judge and we obtain its final score by simply taking the GGN judgment result of it.



(a) Images under ID: 0c4f6bn

(b) Four kinds of garbage classes

Figure 4: Examples of signals and garbage classes. (a) Images of a randomly selected identity from MillionCelebs. (b) Four kinds of garbage classes: fake faces, unrecognizable faces, blurred faces, and face-like patterns.

Multi-task Learning. In order to let LGN learn to identify useless images from different perspectives, a multi-task learning framework is designed. The node classification task predicts scores for the nodes, and the graph classification task predicts scores for the graph to refuse "garbage class" noise. The two tasks promote each other, which allows LGN to better distinguish the difference between outliers and garbages, so it can improve the recognition of both types of noise. "Garbage class" noise is inevitable in face datasets. Images in a garbage class are all wrongly accepted by the face detector and have nothing to help in learning human faces. Figure 4 shows a good class and four kinds of garbage classes in our dataset. With the supervision of the graph classification task, LGN can directly refuse the entire graph if it is judged as a garbage class. All graph classification results of LGNs vote to make the final decision. GGN is not designed as multi-task learning because its big intra-class noise can interfere with the graph classification judgment. In back-propagation, the LGN loss $\mathcal{L}_L$ is a linear sum of two binary cross-entropy loss by a weight term λ: node classification loss $\mathcal{L}_n$ and graph classification loss $\mathcal{L}_g$.

$$\mathcal{L}_L = \mathcal{L}_n + \lambda\mathcal{L}_g \quad (6)$$

$\mathcal{L}_n$ supervises the node predictions like $\mathcal{L}_G$. Differently, to calculate $\mathcal{L}_n$, low confidence nodes are given more weights so that LGN can focus on hard local information. $\mathcal{L}_g$ supervises the network prediction of garbage classes. To calculate it, output features of the second to the last layer of all nodes predicted as signals are averaged as the graph feature. It passes through a fully connected layer with sigmoid activation to obtain the garbage class prediction score.

3.4. Discussion

In order to unify both global and local scales, an end-to-end "Cooperative Learning" (CL) scheme is designed as shown in Figure 5. For a graph data batch, one CL iteration

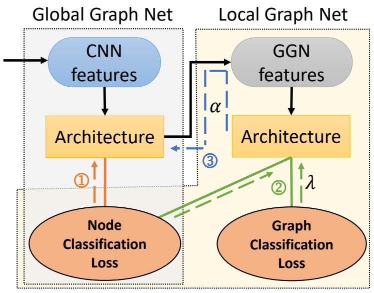


Figure 5: The three-step training scheme "Cooperative Learning". The solid line represents forward propagation and the dotted line represents back propagation.

includes three learning steps. First, update GGN with $\mathcal{L}_G$. Then take output features of the second to the last layer of GGN as the input of LGN, and LGN is updated with $\mathcal{L}_L$. Finally, the gradient of $\mathcal{L}_L$ is multiplied by a scaling factor α to back-propagate to GGN, and GGN is updated again. CL helps the two networks promote each other: In feed-forward, LGN can learn local predictions from easy to hard based on the real-time classification results of GGN to help convergence, and in back-propagation, GGN can improve its global predictions with gradients from hard local regions.

The intra-class label noise is cleansed, but the label flip noise still exists, *i.e.*, face images in one class may actually belong to the person of another class, or two classes may contain face images of the same person. To solve these problems, we average the features of cleansed images in each class to get its feature center. Two classes whose center distance is less than a threshold are combined into one class. Then we compare features of all discarded images with all centers, and put one image into one class if it has a high similarity with the center of that class. Finally, following Cao *et al.* [12], we use the VLAD descriptor clustering [6, 27] to remove duplicated images in the dataset.

4. Experiments

4.1. Experimental Setup

Evaluation Metrics The label noise of data is categorized as outliers, label flips, and garbage classes. In order to evaluate the data cleansing performance, we build a labeled simulation dataset by randomly selecting 2,000 identities for training and 2,000 identities for testing from IMDb-Face [43], which is a manually cleansed face recognition dataset with noise level under 2%. The clustering metrics BCubed precision (P), recall (R) and F1-measure (F) [4] are adapted for the cleansing tasks. Unlike clustering tasks that every sample should be assigned with a specific classification label, outliers in the cleansing task do not belong to any class and should be discarded. So we do not take outliers and samples mistaken as outliers into the iteration of accumulating P and R. Alternatively, signal rate (SR) and the number of images remained in the cleansed dataset (# remained) are calculated to measure the ability to identify and remove outliers. For the real data validation, we evaluate face recognition performance of ResNet [23] models trained on original and cleansed datasets by the Arcface loss [15]. Ten-fold verification sets [12, 17, 26, 35, 40, 55, 56] are used to test face verification accuracy. The MegaFace Challenge 1 [28] evaluates face recognition performance under 1M distractors environment. The IJB benchmarks [34, 46] evaluate template-wise face recognition performance.

Datasets	# photos	# subjects	Noisy
CASIA-WebFace [54]	0.5M	10K	×
IMDb-Face [43]	1.7M	59K	×
VGGFace2 [12]	3.3M	9K	×
MegaFace2 [36]	4.7M	672K	✓
MS-Celeb-1M [21]	7.5M	100K	✓
MillionCelebs	87.0M	1M	✓
- MegaFace2 [36]	20.0M	734K	×
- FaceGraph	18.8M	636K	×

Table 1: Training datasets used in our experiments. "✓" in the last column means label noise rate $>$ 30%.

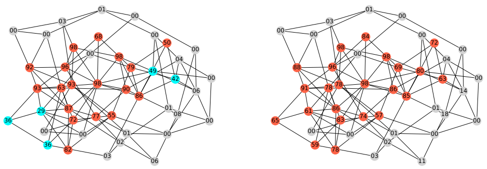
Implementation Details Table 1 shows face training sets used in our experiments, where "-X" means the dataset is cleansed by method "X". To guarantee the reliability of graph structures, we obtain 512-dimensional face features from a ResNet100 Arcface model trained with cleansed MS1M, then build 3-NN graphs with self-loop on all nodes. GCNs are designed as 5 layers with 256-dimensional hidden features. The learning rate is 0.001 with weight decay 0.0005 and graph batch size 50. α and λ are set 1 and 0.5.

4.2. Experiments on Simulation Datasets

In this section, we add noise to the simulation dataset and re-cleanse it: We gradually replace its images with randomly selected images from the rest of IMDb-Face as outliers, and randomly put images from one identity to another as label flips. Besides, the simulation dataset is always polluted by 10% "garbage class" noise selected from MS1M [21]. Comparative methods include method of Bansel *et al.* [7], MegaFace2 [36], VGGFace2 [12] and Celeb500k [11]. GCN [29] and GraphSAGE [22] are also trained in the same way as FaceGraph. An ablation study is made with four setups: only GGN, separately trained FaceGraph (GGN + LGN), and CL trained FaceGraph with (CL + MT) or without (CL) multi-task learning. In the absence of multi-task learning, networks are trained to predict all nodes as noise for garbage classes. For a fair comparison, the garbage removal strategy of MegaFace2 [36] is applied

Cleansing Methods	F (%)	SR (%)	# remained
Noisy	-	-	54436
Bansel *et al.* [7]	83.30	77.10	30128
VGGFace2 [12]	48.04	41.43	35481
Celeb500k [11]	73.07	72.29	17187
MegaFace2 [36]	82.67	81.67	26711
GCN [29]	85.31	87.06	20616
GraphSAGE [22]	85.59	83.26	26985
GGN	89.09	81.99	28660
FaceGraph - GGN+LGN	89.86	78.63	**29053**
FaceGraph - CL	89.47	81.97	28729
FaceGraph - CL+MT	**90.03**	**95.59**	24071

Table 2: F-score, signal rate and the number of remained images of different methods under garbage rate 10%, outlier rate 30%, and label flip rate 30%.

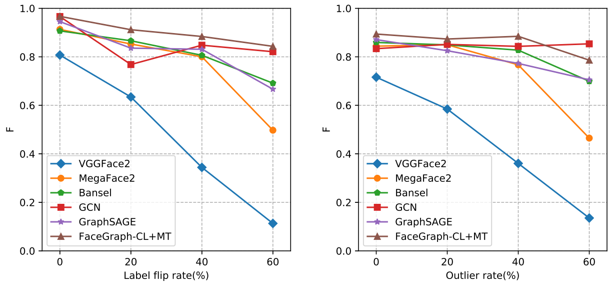


(a) GGN (b) GGN + LGN

Figure 6: FaceGraph output graph of one class in the test set. The numbers on nodes are the prediction scores of networks (in percentage). Wrongly judged nodes (blue) of GGN are corrected by the second-stage LGN, so all noise (gray) and signals (red) are successfully classified.

to methods that do not consider garbage classes.

In Table 2, we cleanse a dataset with 30% outliers and 30% label flips. It is observed that the GCN-based methods remain stronger discrimination ability while comparable methods lose power in a big noise environment. For FaceGraph, when we separately train two-stage networks, it performs 3.36% worse than a single GGN in signal rate, meaning LGN cannot learn local details efficiently. However, when trained with "CL", FaceGraph maintains the same level signal rate with more images remained than a single GGN, and reaches a higher F at 89.47%. To explore how LGN affects judgment, we visualize the output graph of one class in test sets as in Figure 6. We find that the low confidence nodes, especially wrong nodes, tend to gather together to form some "local regions" that are difficult to deal with. Our algorithm reasonably builds more subgraphs in the difficult regions, so the final judgment of the nodes in these regions can fuse more opinions. It is also observed that the wrong nodes in GGN usually occur in large complicated graphs, while a two-hop range maintains most information for valid graph convolution, so LGNs can focus on the data distribution of local and hard cases by receiving only small subgraphs as input to simplify the prediction.



(a) F varies with label flip rate (b) F varies with outlier rate

Figure 7: Model robustness at different noise levels. The FaceGraph model is stable with the noise rate changing.

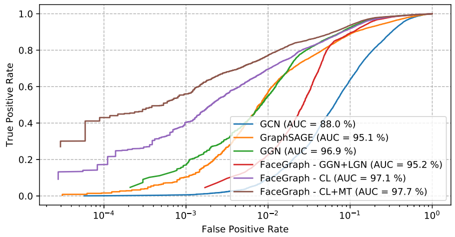


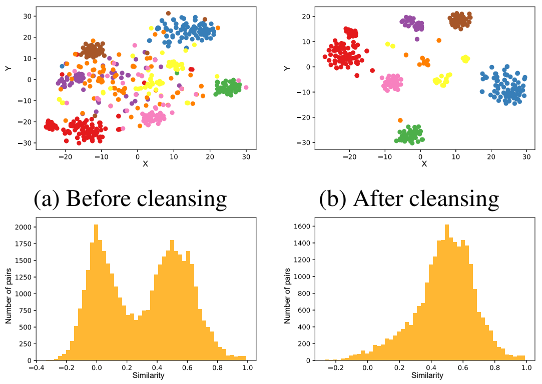
Figure 8: ROC curves and AUCs of recognizing useless images. The FaceGraph models can discard useless images with higher confidence than other GCN-based methods.

Figure 7 shows cleansing the dataset with gradually contaminated noise. In 7a, we set the outliers at 30%, and vary the label flips from 0% to 60% in steps of 20%; In 7b, we set the label flips at 30%, and vary the outliers from 0% to 60% in steps of 20%. Celeb500k's model [11] is excluded from comparison because it is trained for specific classes without generalization ability. We find that algorithms have similar performance when the noise rate is low. However, artificial designed methods, especially SVM trained VGGFace2's method [12], drop sharply at high noise rate, while the FaceGraph model is robust to different noise levels.

ROC curves in Figure 8 compare the ability to remove useless images (outliers and garbage classes) of GCN-based methods. For calculation convenience, we adjust outliers to 60% and do not add label flips. FaceGraph methods all reach higher AUC than GCN and GraphSAGE, for example, a single GGN model surpasses GCN [29] by 8.9% and GraphSAGE [22] by 1.8%. Except "GGN + LGN", FaceGraph methods have much higher TPR at FPR less than 1e-3, which means that they are more confident to remove obvious outliers and garbages. Under the same circumstance, "CL" outperforms "GGN + LGN" by 1.9% AUC. "CL + MT" achieves the highest F (90.03%), SR (95.59%) and

Training Datasets	LFW	CFP-FP	AgeDB	CALFW	CPLFW	SLLFW	VGG2-FP	Average
CASIA [54]	99.38	95.19	94.55	92.28	85.90	**98.17**	93.38	94.12
-Cleansed by [3]	99.30	95.01	94.45	92.15	86.22	97.93	93.18	94.03
-FaceGraph	**99.42**	**95.19**	**94.65**	**92.55**	**86.43**	97.88	**93.78**	**94.27**
VGGFace2 [12]	99.58	96.93	95.73	**93.63**	92.07	98.87	95.80	96.08
-FaceGraph	**99.62**	**97.03**	**95.98**	93.53	**92.13**	**98.88**	**96.04**	**96.17**
MegaFace2 [36]	99.57	91.67	89.40	**89.82**	83.52	98.13	91.98	92.01
-FaceGraph	**99.58**	**92.93**	**89.80**	89.15	**84.92**	**98.32**	**92.46**	**92.45**
MS1M [21]	99.60	94.16	96.40	93.06	86.83	98.98	93.70	94.67
-IBUG [16]	99.80	92.76	97.70	95.35	87.45	99.48	93.04	95.08
-DeepGlint [1]	99.80	93.66	97.82	95.63	88.75	99.43	92.16	95.32
-FaceGraph	**99.80**	**96.90**	**97.92**	**95.67**	**92.27**	**99.50**	**95.42**	**96.78**

Table 3: Cleanse 4 face recognition datasets and train deep models by Arcface [15] to test face verification accuracy (%). FaceGraph cleansed versions surpass others on at least 6 out of 7 verification sets, and always enhance the average accuracy.



(a) Before cleansing

(b) After cleansing

(c) Before cleansing

(d) After cleansing

Figure 9: Visualization of cleansing randomly selected 8 classes from MS1M. (a)(b) t-SNE [31] data distribution; (c)(d) Histogram of pairwise cosine similarity in one class.

AUC (97.7%) while keeping the number of remained images moderate, showing that multi-task learning not only helps to recognize garbage classes but also helps to cleanse outliers and label flips, because the model can better distinguish the difference between different types of noise.

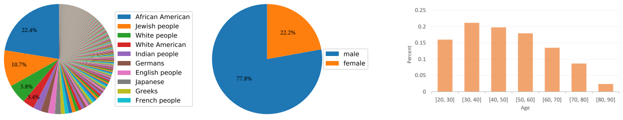
4.3. Experiments on Existing Datasets

A noise-free dataset can significantly enhance the face recognition performance [7, 20, 43], but existing face recognition datasets mostly suffer from label noise problem [43]. In this section, four widely used face datasets are cleansed by the "FaceGraph - CL+MT" model. Considering the data size, CASIA [54] is trained with ResNet34 [23], while others are trained with ResNet50 [23]. Identities that contain less than 8 images in the MegaFace2 dataset [36] are deleted because too many identities with a small number of images can affect its convergence. Table 3 compares face verification performances of the original and cleansed datasets. With a smaller amount of data, FaceGraph achieves the highest average accuracy for all datasets and performs better on at least 6 out of 7 test sets. For a dataset with big noise like MS1M [21], FaceGraph gets significant improvement on all test sets and enhances the average recognition accuracy by 2.11%. On the cross-pose and cross-age tests, MS1M-FaceGraph outperforms MS1M by 2.61% on CALFW [56] and 5.44% on CPLFW [55], showing that FaceGraph can master large pose and age gap cases, reserving variations of the same person. Therefore, FaceGraph makes a good trade-off between cleanliness and diversity.

Randomly selecting 8 classes from MS1M [21], Figure 9 visualizes the feature space distribution using t-SNE [31] embeddings and the histogram of intra-class pairwise similarity of images before and after cleansing. In Figure 9a9b, every class is represented by one color. The original data distribution is very scattered. After cleansing, most scattered samples in one class are discarded, leaving one group that can gather together. In Figure 9c9d, there are two main peaks in the histogram before cleansing, which are at similarity 0.0 and 0.5. The former is obviously caused by the label noise. After cleansing, the peak at 0.0 disappears, meaning that the noise is removed successfully. Moreover, FaceGraph does not eliminate all low-similarity pairs. There are still a few pairs around 0.0 to reserve face variation.

4.4. Experiments on Large-Scale Cleansing

MillionCelebs A large-scale face recognition training set, namely MillionCelebs, is collected according to a celebrity name list released by Guo *et al.* [21]. We download 50-100 images for each celebrity from the Internet Image Search Engine, detect faces with MTCNN [50], then align and crop the images to 112×112 face warps. In this way, we get 87.0M face images of 1M identities. FaceGraph is used to cleanse these images, then we carefully remove



(a) Ethnicity (b) Gender (c) Age

Figure 10: Demography statistics of MC-FaceGraph.

identities overlapping with LFW [26], FaceScrub [37] and IJB [34, 46]. Finally, we obtain a cleansed version with 18.8M faces of 636.2k identities, namely MC-FaceGraph. Figure 10 shows demography statistics of MC-FaceGraph extracted from Freebase [2]. It has a great variety in the distribution of human ethnicity and age. Most importantly, it is a large-scale dataset with very high cleanliness, which can significantly improve the face recognition performance.

To verify the value of FaceGraph in large-scale applications, a ResNet100 [23] face recognition model is trained with MC-FaceGraph. Training with initial SGD learning rate 0.1, weight decay 0.0005 and batch size 720, we decrease the learning rate by 0.1 at the 366,000th, 498,000th, 638,000th iterations, and stop at the 748,000th iteration. Comparable models are trained under the same environment with different training sets. For a fair comparison, the cleansing method of MegaFace2 [36] is also applied to cleanse MillionCelebs, noted as "MC-MegaFace2" [36].

MegaFace MegaFace Challenge 1 [28] is a large-scale face recognition challenge that tests the model performance under one million distractors. It measures TPR at 1e-6 FPR for verification and Rank-1 retrieval performance for identification. In Table 4, adopting FaceScrub [37] as probe set and using the wash list provided by DeepInsight [15], the results of two MillionCelebs cleansed versions do not differ a lot, but they all outperform other training datasets by a large margin. For instance, the identification accuracy of MC-FaceGraph is 0.67% higher than MS1M-V2 [15] to reach 99.02%. It nearly saturates the MegaFace Challenge on both identification and verification protocols.

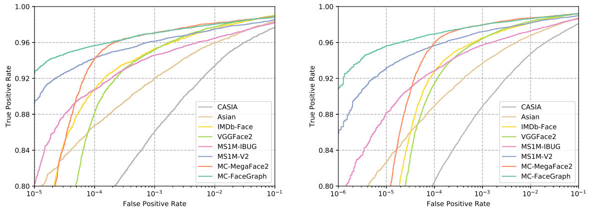
IJB The IJB-B [46] and IJB-C [34] benchmarks test template-wise face recognition performance. The verification TPR at 1e-5 FPR and identification Rank-1 are reported in Table 5. MC-FaceGraph trained model surpasses all candidates by a large margin. Figure 11 compares ROC curves of listed methods. Two MillionCelebs versions have the same level performance at higher FPR. However, verification accuracy of the MC-MegaFace2 [36] drops sharply at 1e-4 FPR, and becomes worse than many small-scale datasets at 1e-5 FPR. This shows that a large number of identities and images do not necessarily mean an increase in face recognition performance. If there are strict requirements for identifying negative pairs, it is essential to train with a dataset of great cleanliness to learn detailed features.

Training Datasets	Ver.(%)	Id.(%)
CASIA [54]	97.11	92.93
Asian [1]	94.90	91.21
IMDb-Face [43]	97.87	96.26
VGGFace2 [12]	98.00	95.54
MS1M-IBUG [16]	98.25	97.53
MS1M-V2 [15]	98.48	98.35
MC - MegaFace2 [36]	**98.97**	98.96
MC - FaceGraph	98.94	**99.02**

Table 4: Verification TPR (@FPR=1e-6) and identification Rank-1 on the MegaFace Challenge 1 [28]. "MC-X" means MillionCelebs cleansed by method "X".

Training Datasets	IJB-B		IJB-C	
	Ver.(%)	Id.(%)	Ver.(%)	Id.(%)
CASIA [54]	62.42	86.70	69.61	88.05
Asian [1]	79.12	91.29	82.64	92.26
IMDb-Face [43]	64.87	93.41	66.85	94.52
VGGFace2 [12]	41.64	93.20	59.33	94.44
MS1M-IBUG [16]	80.27	92.19	88.16	93.54
MS1M-V2 [15]	89.33	94.50	93.15	95.72
MC - MegaFace2 [36]	62.67	95.04	76.29	96.10
MC - FaceGraph	**92.82**	**95.76**	**95.62**	**96.93**

Table 5: Verification TPR (@FPR=1e-5) and identification Rank-1 on the IJB-B [46] and IJB-C [34] benchmarks.



(a) IJB-B (b) IJB-C

Figure 11: Verification ROC curves on IJB-B and IJB-C. MC-FaceGraph keeps good performance at very strict FPR.

5. Conclusion

In this paper, we propose a novel label noise cleansing method FaceGraph and build a large-scale face recognition dataset MillionCelebs. In the experiments, FaceGraph provides high-quality cleansing results, surpassing existing methods in the ability to find and reject label noise. The MillionCelebs dataset cleansed by FaceGraph also achieves remarkable performance on many benchmarks.

Acknowledgments This work was supported by Canon Information Technology (Beijing) Co., Ltd. under Grant No. OLA19023.

References

[1] Challenge 3: Face feature test/trillion pairs. `trillionpairs.deepglint.com`.

[2] Freebase data dump. `www.freebase.com`.

[3] Github: happynear/faceverification. `http://github.com/happynear/FaceVerification/`.

[4] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[5] Anelia Angelova, Yaser Abu-Mostafam, and Pietro Perona. Pruning training sets for learning of object categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 494–501. IEEE, 2005.

[6] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[7] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. The do's and don'ts for cnn-based face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2545–2554, 2017.

[8] Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 464–473. IEEE, 2017.

[9] Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999.

[10] Carla E Brodley, Mark A Friedl, et al. Identifying and eliminating mislabeled training instances. In *Proceedings of the National Conference on Artificial Intelligence*, pages 799–805, 1996.

[11] Jiajiong Cao, Yingming Li, and Zhongfei Zhang. Celeb-500k: A large training dataset for face recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2406–2410. IEEE, 2018.

[12] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Automatic Face &amp; Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 67–74. IEEE, 2018.

[13] Zhengdao Chen, Xiang Li, and Joan Bruna. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*, 2017.

[14] Sarah Jane Delany, Nicola Segata, and Brian Mac Namee. Profiling instances in noise reduction. *Knowledge-Based Systems*, 31:28–40, 2012.

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

[16] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. Marginal loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 60–68, 2017.

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

[18] Weiwei Du and Kiichi Urahama. Error-correcting semi-supervised learning with mode-filter on graphs. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2095–2100. IEEE, 2009.

[19] Weiwei Du and Kiichi Urahama. Error-correcting semi-supervised pattern recognition with mode filter on graphs. In *2010 2nd International Symposium on Aware Computing*, pages 6–11. IEEE, 2010.

[20] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

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

[22] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Mauricio A Hernández and Salvatore J Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data mining and knowledge discovery*, 2(1):9–37, 1998.

[25] Ray J Hickey. Noise modelling and evaluating learning from examples. *Artificial Intelligence*, 82(1-2):157–179, 1996.

[26] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[27] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011.

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

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] Stéphane Lallich, Fabrice Muhlenbach, and Djamel A Zighed. Improving classification by removing or relabeling mislabeled instances. In *International Symposium on Methodologies for Intelligent Systems*, pages 5–15. Springer, 2002.

[31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[32] Jonathan I Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Iq*, pages 200–209. Citeseer, 2000.

[33] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.

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

[35] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *Computer Vision and Pattern Recognition Workshops*, pages 1997–2005, 2017.

[36] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3406–3415. IEEE, 2017.

[37] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*, pages 343–347. IEEE, 2014.

[38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[39] P Jonathon Phillips, Amy N Yates, Ying Hu, Carina A Hahn, Eilidh Noyes, Kelsey Jackson, Jacqueline G Cavazos, Géraldine Jeckeln, Rajeev Ranjan, Swami Sankaranarayanan, et al. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences*, 115(24):6171–6176, 2018.

[40] C.D. Castillo V.M. Patel R. Chellappa D.W. Jacobs S. Sengupta, J.C. Cheng. Frontal to profile face verification in the wild. In *IEEE Conference on Applications of Computer Vision*, February 2016.

[41] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[42] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[43] 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. In *European Conference on Computer Vision*, pages 780–795. Springer, 2018.

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

[45] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1117–1125, 2019.

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

[47] D Randall Wilson and Tony R Martinez. Instance pruning techniques. In *ICML*, volume 97, pages 400–411, 1997.

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

[49] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[50] Jia Xiang and Gengming Zhu. Joint face detection and facial expression recognition with mtcnn. In *Information Science and Control Engineering (ICISCE), 2017 4th International Conference on*, pages 424–427. IEEE, 2017.

[51] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[52] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. pages 2298–2306, 2019.

[53] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

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

[55] 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.

[56] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments. *CoRR*, abs/1708.08197, 2017.

[57] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1237–1246, 2019.

[58] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7812–7821, 2019.

[59] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Learning to adapt invariance in memory for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.