IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 2, 2022, accepted January 8, 2022, date of publication January 13, 2022, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142922

Confidence-Based Simple Graph Convolutional Networks for Face Clustering

DENGDI SUN[1,2], KANG YANG[2], AND ZHUANLIAN DING[3]

[1]Key Laboratory of Intelligent Computing and Signal Processing (ICSP), Ministry of Education, School of Artificial Intelligence, Anhui University, Hefei 230601, China
[2]Anhui Provincial Key Laboratory of Multimodal Cognitive Computing, School of Computer Science and Technology, Anhui University, Hefei 230601, China
[3]School of Internet, Anhui University, Hefei 230039, China

Corresponding author: Zhuanlian Ding (dingzhuanlian@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61906002, Grant 62076005, and Grant U20A20398; in part by the Natural Science Foundation of Anhui Province under Grant 2008085MF191, Grant 2008085QF306, and Grant 2008085UD07; and in part by the University Synergy Innovation Program of Anhui Province, China, under Grant GXXT-2021-002.

ABSTRACT Face clustering is an effective method for taking advantage of unlabeled face data. Recent studies use graph convolutional networks (GCNs) to learn feature embeddings from the neighborhood information between face images. However, most of the face clustering methods require numerous overlapping subgraphs to characterize the local structure around the nodes, which causes significant redundancy. Moreover, the nonlinearity of the GCN itself increases the calculation complexity, which further reduces the model's training efficiency. In this study, we propose a lightweight clustering framework, the confidence-based simple graph convolutional network (CSGCN), for face clustering, which achieves more accurate clustering results and significantly improves the efficiency of GCN-based face clustering. Specifically, CSGCN does not construct any subgraphs but convolves the entire graph as a whole and also removes the nonlinearity of the convolution in the graph convolution module, which further reduces the computational complexity. Subsequently, an effective new confidence score is constructed to better characterize the embedded features and to ensure that the subsequent clustering still maintains a high accuracy rate under the aforementioned model simplification. In addition, while most of the existing GCN-based methods are actually supervised, we construct an unsupervised confidence to make it more suitable for clustering tasks. Extensive experiments with MS-Celeb-1M, YouTube-Faces and DeepFashion datasets show that our method not only improves the clustering accuracy but also significantly reduces the execution time, whether in supervised or unsupervised models.

INDEX TERMS Face clustering, confidence score, simple graph convolutional networks.

I. INTRODUCTION

In recent years, with the progress of face detection technology, large numbers of face images are easily obtainable from various surveillance cameras and through the internet. However, labeling these face images is very time-consuming and expensive. Datasets, manually labeled by humans, are of questionable accuracy [1]. Automatically analyzing facial features is a necessary advancement. Face clustering is an effective and basic task in facial feature analysis with a wide range of applications, and has been extensively studied in previous works [2]–[14]. There are many important applications for face clustering in image retrieval [5], [15], data cleaning and marking [9], [16]–[19], and criminal investigation [20].

Some traditional clustering methods, such as the K-means clustering algorithm [21], density-based spatial clustering of applications with noise (DBSCAN) [22], and hierarchical agglomerative clustering (HAC) [23], [24], make unrealistic assumptions about the distribution of data. The drawbacks of these algorithms limit their application in face clustering problems with complex distributions of facial representations. To address this constellation of real problems, recent studies have shown that utilizing graph convolutional networks (GCNs) and supervised information [7], [8], [10], [13] can enhance the characteristics of face clustering. GCNs learn cluster patterns rather than completing a cluster.

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

Although these graph-convolution-based supervised methods can effectively deal with complex clustering patterns, they also have some problems. First, they construct numerous overlapping subgraphs [7], [10]. The subgraphs are overly redundant, which increases the consumption of computing resources and severely limits their efficiency. Second, these methods [7], [8] make assumptions when constructing subgraphs, and need to set many hyperparameters, which increases the difficulty of parameter adjustment and model training. Third, the nonlinear calculation complexity of the GCN itself is high, which affects the training time. Thus, fast and accurate clustering is still difficult for face clustering. In response to these problems, we do not attempt to construct subgraphs, but to operate on the entire graph, with the intention of avoiding the assumption of constructing subgraphs and the adjustment of hyperparameters. Additionally, we remove the nonlinearity of the GCN, making our model more efficient. To maintain a good clustering effect in a simplified situation, we construct an effective confidence score to guide the subsequent clustering module.

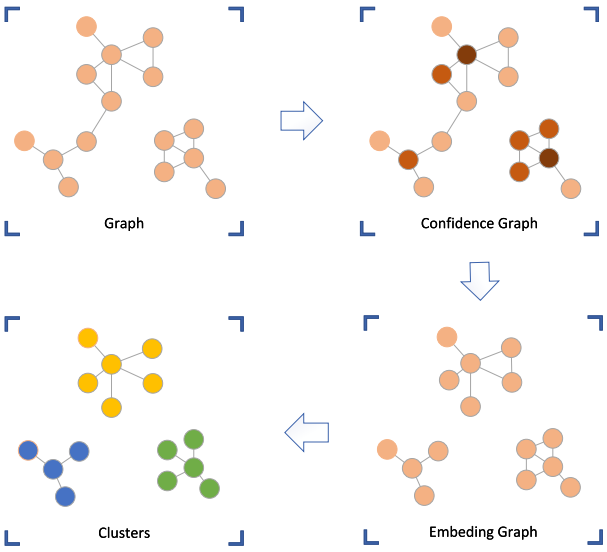


FIGURE 1. The core idea of our model. The confidence of the nodes in the graph is learned through the graph convolution module. The new graph is reconstructed from the obtained embedding feature information. Then the reconstructed graph is combined with the confidence to obtain the final clustering result through the clustering module.

Specifically, we propose a new face clustering framework called a confidence-based simple graph convolutional network (CSGCN). Fig 1 shows the core idea of our model. Previously, we converted the face feature data extracted by convolutional neural networks (CNNs) into graph data using K nearest neighbor (KNN) method. We design two modules for our framework, namely, the graph convolution module and the clustering module. The graph convolution module embeds the original data and obtains the confidence of each node. The confidence score indicates the possibility of a node belonging to a certain category. The clustering module is used to group the nodes into clusters according to the confidence scores and embedding features. In summary, unlike the other face clustering models, CSGCN is more efficient and works well without labels.

Finally, we conduct numerous experiments on the MS-Celeb-1M [16] and YouTube-Faces [25] datasets. The results of those experiments show the advantages of our CSGCN model. In addition, we also test it on a subset of a challenging long-tailed dataset called DeepFashion [26] with similar results. The main research contributions of this study are as follows:

- We propose a new face clustering framework that does not construct any subgraphs but convolves the entire graph as a whole and removes the nonlinearity of the convolution in GCN, which significantly improves the efficiency.
- Most of the existing GCN-based methods are supervised, whereas our method can be extended to an unsupervised version, which is more suitable for clustering tasks.
- The proposed method shows good performance and efficiency for the MS-Celeb-1M, YouTube-Faces and DeepFashion datasets.

In this section, we briefly introduce our research. The rest of this paper is structured as follows: In Section II, we summarize related studies of face clustering and GCN in detail. Section III presents the specific implementation details of the model proposed in this article. Section IV presents a series of experiments to analyze the performance of various clustering algorithms. Finally, we summarize this study and point towards future work in Section V.

II. RELATED WORK

In this section, we will introduce related works on face clustering and graph convolutional neural networks.

A. FACE CLUSTERING

Face clustering is a method used to process large amounts of unlabeled face data. However, the face data extend across a large-scale, and the facial feature distribution is very complicated. Some conventional and common clustering algorithms show poor performance in real and complex face clustering tasks. K-means [21] needs to set the value of k in advance, and the clustering result is highly affected by the value of k. The time complexity of the spectral clustering [27] is extremely high, which is not conducive to expanding to large datasets. DBSCAN [22] and hierarchical DBSCAN [28] produce poor clustering effects on high-dimensional data. Border-peeling clustering [29] produces excessive clustering, forming too many clusters for data with complex shapes. Robust border-peeling clustering [30] and robust continuous clustering [31] cannot effectively handle high dimensions and large datasets. Overall, these traditional clustering algorithms make some unrealistic assumptions about the distribution of data. Thus, face clustering remains a challenging task.

Early face clustering methods focused on the design of handcrafted features [15], [32] and then used traditional

clustering algorithms to cluster faces. Owing to progressions in deep learning, subsequent works make use of deep features and concentrate on the design of similarity metrics. Zhu *et al.* [5] proposed a method called the rank-order distance to measure the affinity between two face images. Lin *et al.* [33] trained a linear support vector machine (SVM) based on the nearest neighbors of face samples to calculate the similarity measure between deep facial features. Otto *et al.* [3] proposed a face clustering method based on an approximate rank-order to measure face image pairs. Shi *et al.* [4] designed conditional pairwise clustering(ConPaC), which formulated the clustering task as a conditional random field model. Lin *et al.* [2] designed a new density-based strategy by introducing minimal covering spheres of neighborhoods based on support vector data descriptions (SVDDs) [34]. Zhan *et al.* [9] trained a classifier to aggregate multiview information to select face image pairs that belong to the same category.

Compared with the above studies, GCN is an effective method for processing graph structure data so it can be better applied to face clustering problems. Recent studies have shown that introducing supervisory information into face clustering can improve the performance. Wang *et al.* [7] transformed the face clustering problem into a link prediction problem. This was the first study to apply the GCN to the task of face clustering. They trained a GCN model to predict the link probability between the pivot node and its neighboring nodes. Yang *et al.* [8] learned clustering patterns by detecting segmentation paradigms. A GCN-based detection and segmentation module was proposed to complete the face clustering task. Yang *et al.* [10] used GCN to infer the confidence of nodes and the connectivity of edges to complete clustering. Guo *et al.* [11] fused GCN and long short-term memory (LSTM) to obtain embedded face data based on density and then used traditional algorithms to achieve a good effect.

B. GRAPH CONVOLUTIONAL NETWORK

Convolutional neural networks (CNNs) have been widely used in many fields such as image classification. The data in these fields are usually presented in the form of a regular grid in Euclidean space. However, many non-Euclidean structural data exist naturally in real applications, such as social analysis [35], [36] and computer vision [7], [8], [37]. Because of the diversity and complexity of the graphic structure, CNNs cannot process it directly. Graph convolutional networks (GCNs) [38] are a natural extension of CNNs in the graph domain and are used to explore the relationship and interdependence between objects in the graph. GCNs have proven to perform well in many tasks [38]–[42]. Some recent studies extended GCN to deal with large-scale graphs. Hamilton *et al.* [39] used a multilayer aggregation function to sample the neighbors in each layer, without any dependencies on the global graph structure. Chen *et al.* [43] further reduced the computational costs by sampling nodes instead of neighbors. The model proposed by Wu *et al.* [44] was easy to train and is easily extended to large datasets. Experiments show that simple graph convolutional networks (SGCNs) will not have a significant negative impact on the accuracy of many graph-based tasks but they will bring great efficiency improvements.

III. METHODOLOGY

In this section, we elaborate on our proposed model framework, including the graph convolution module and clustering module. We will introduce in detail the design of the supervised and unsupervised confidence in the model as well as a time complexity analysis.

A. OVERVIEW

Clustering is performed on the given face feature data $X = [x_1, \cdots, x_N]^T \in \mathbb{R}^{N \times D}$, where N is the number of face images, and D is the dimension of the data feature. Our main goal is to assign the same label y_i to multiple face pictures of the same person.

Our mission is to find valuable face pictures in complex data. The pictures are equivalent to cluster centers. However, what kind of picture is a face photo with important information? In a real face dataset, the face photos of the same person will have various postures, angles, expressions and possibly occluded parts, or the environment behind them may be messy. The faces that were captured from a frontal angle with proper brightness and a normal expression will be the faces with significant information; these are called pivot faces. These face pictures are the most valuable faces that we need. We believe that the cluster centers of clusters are generally these pivot face images, and these images belong to a certain class with high credibility.

The task of the first stage of our model is to determine the pivot face images. The next task is to link the other face images to the pivot face images. The connected subgraphs form a cluster. We map the relationship between face images to the relationship between the nodes in the graph. The relationship between the face images is similar to the structural information between nodes in the graph. We hope that structural information can be learned using a learnable model. Graph convolution is a good choice for learning the structural information of graphs by aggregating adjacent information.

As shown in Fig 2, our framework is divided into two modules. The graph convolution module utilizes the GCN module to automatically learn to find the pivot faces, which are the cluster centers in the cluster. The clustering module uses a clustering algorithm to obtain the connected graph through the relationship between the cluster center and the neighbors to complete the clustering task.

B. GRAPH CONVOLUTION MODULE

First, to obtain a pivot face image, we design a model based on the GCN. It is primarily used to predict the probability that face images are pivot faces. Secondarily, it is used to obtain a better feature embedding representation.

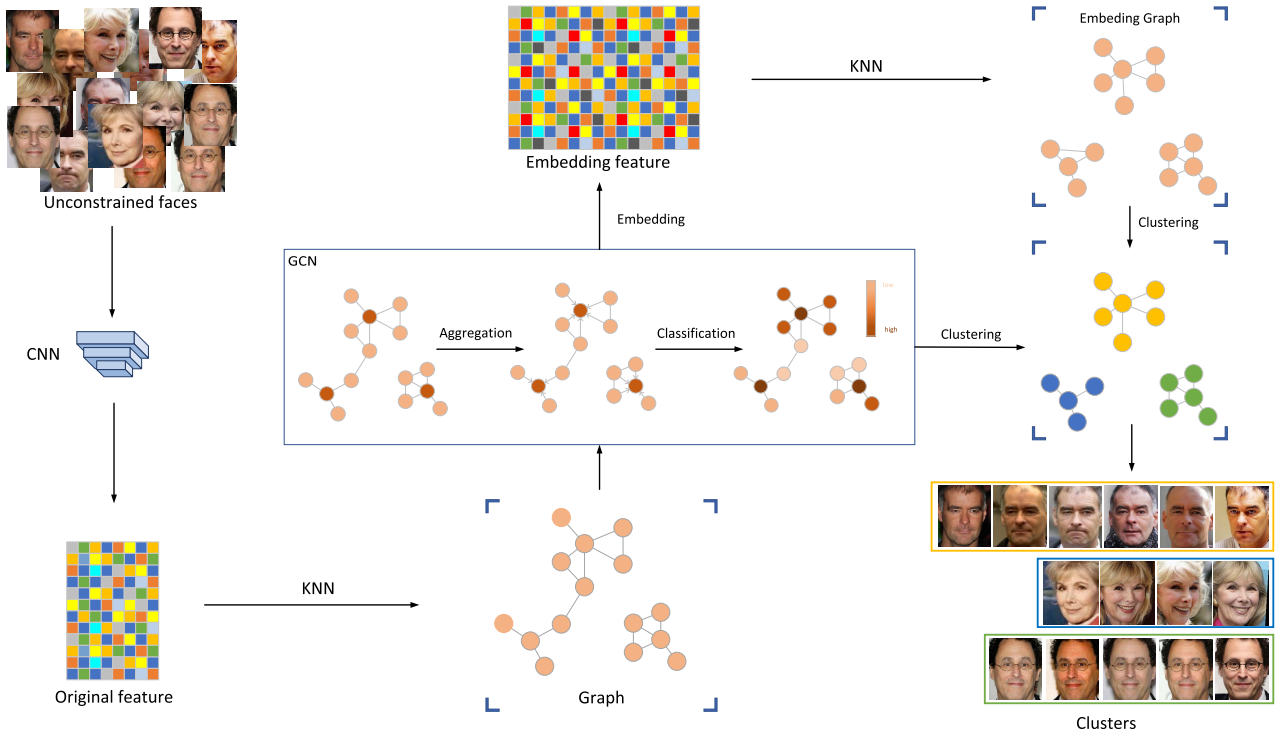


FIGURE 2. Overview of the proposed CSGCN clustering framework. The original image is passed through CNN to obtain the embedded features, and then the embedded features are used to form a graph through KNN. We put the graph into the graph convolution module, obtain new node embedding features through aggregation, and obtain the confidence score of the node through classification. Then, we recompose the node embedding feature to obtain a new embedding graph and put the feature embedding graph and the node confidence set into the clustering module to get the clustering result.

Each face image is regarded as a node, and $G = (V, E)$ is obtained through the KNN composition. We convert the face feature data into graph data. The clustering of face images is converted to the clustering of nodes in the graph. The adjacency matrix of the graph is expressed as $A \in \mathbb{R}^{N\times N}$, where $a_{i,j}$ is the affinity relation between nodes i and j, which is the cosine similarity between x_i and x_j. If the two nodes are not connected, then $a_{i,j} = 0$.

The design of our GCN model is as follows: the model takes the feature embedding matrix as input:

$$\mathbf{F}_0 = \mathbf{X}.$$

According to some theoretical experience of SGCN [44], we make the embedding matrix F_{l+1} of the $l+1$ layer a linear aggregation of layer l; therefore, at each step, the embedding matrix is defined as:

$$\mathbf{F}_{l+1} = \alpha\mathbf{F}_l + (1-\alpha)\mathbf{D}^{-1}\mathbf{A}\mathbf{F}_l\mathbf{W}_l,$$

where A is the adjacency matrix of nodes in the graph, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ is the diagonal matrix, α is a balance parameter, which is learnable and used to balance the importance of the weight of the updated feature and its own feature. $\mathbf{W}_l$ is a parameter matrix that can be learned and used to transform the feature embedding. This is similar to the parameter matrix in CNN, except that it is used for a graph structure.

The GCN model aggregates the node features and its adjacent node features to obtain a new feature embedding F_L. After embedding the features, we access a node classifier based on a fully connected network. Each node in the graph will obtain a score s', as follows:

$$s' = F_L W + b.$$

We design a confidence score s from the real label data, as follows:

$$s = \lambda \cdot \frac{1}{|N_i|}\left(\sum_{\substack{v_j\in N_i,\\ y_j=y_i}} a_{i,j} - \sum_{\substack{v_i\in N_i\\ y_j\neq y_i}} a_{i,j}\right) + (1-\lambda)\cdot\frac{1}{|S_i|}\sum_{v_j\in S_i} a_{i,j},$$

where N_i represents the neighborhood around v_i, $a_{i,j}$ is the affinity between v_i and v_j, y_i is the ground-truth label of v_i, and S_i is all the vertices with the same label as v_i. This score reflects the importance of nodes. The higher the score, the greater the probability that it is the desired pivot face image.

The former is the relationship between a node and its surrounding neighbors. This reflects whether it is an important central node. The latter is the average similarity between a node and all the nodes with the same label as this node. This reflects a type of global information, which will shorten the distance of the same class of nodes in the GCN aggregation process.

Additionally, we design an unsupervised method to define confidence based on [45], [46]. The unsupervised confidence score is defined as follows:

$$s_u = \frac{1}{|\delta_i|} \sum_{v_j \in \delta_i} e_{i,j}.$$

Given a radius, δ_i is the set of nodes in the range of v_i, and v_j is the node of the set δ_i. $e_{i,j}$ represents the edge weight between v_i and v_j.

Due to the powerful representation abilities of GCN, we use feature embedding after GCN aggregation. This also shows the effect of not losing supervision information in the experiment.

According to this confidence design, the loss function of the node classifier model is as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left| s_i - s_i' \right|^2.$$

The feature embedding obtained through GCNs is also very important. We save the embedding feature F_L obtained from the top layer of the GCN and use KNN to form a new graph. Then, further processing is performed on the new embedding graph to obtain the clustering result.

Our method uses a subgraph-free mode, compressing the entire graph into the GCN. The convolution process aggregates the characteristics of the neighboring nodes around each node each time. Since no special subgraph selection strategy is required, many hyperparameters for selecting subgraphs in many previous studies are omitted. This allows our model to be tuned better and faster than other methods based on the GCN.

C. CLUSTERING MODULE

In the clustering stage, we obtain the feature embedding F_L and the confidence set S. As shown in Algorithm 1, we connect each node to the surrounding high-confidence nodes to obtain a connected relationship. We use the weighted merge search algorithm for path compression to connect the connected graph to form a subgraph. Specifically, we find the root nodes of all the nodes and the nodes with the same root node belonging to the same cluster. We assign them cluster class identifiers to obtain the clustering results.

D. COMPLEXITY ANALYSIS

In the graph convolution module, the computational cost is mainly concentrated in the graph convolution part. Since the value of K in the graph is much smaller than N, Matrix A is a particularly sparse matrix, which means that graph convolution can be implemented as sparse matrix multiplication. The number of edges in Matrix A is E, so the time complexity of the graph convolution is $O(|E|)$. The time complexity of the edge generation part of the clustering module is $O(nlogn)$, and the time complexity of the link edge is $O(logn)$.

Algorithm 1 Confidence-Based Clustering

Input: Face image set V; Embedding feature F; Confidence score set S; Threshold θ; Number of neighbors k;
Output: Clusters *clusterId*;

M = findMostConfidenceNodes(S)
$edges = \phi$
for all m_i in M **do**
  find its k nearset neighbors v_j
  if dist(m_i, v_j) < 1 - θ **then**
    $edges = edges \cup \{\{m_i, v_j\}\}$
  end if
end for
for all (e_i, e_j) in edges **do**
  u = findParent(e_i)
  v = findParent(e_j)
  if size[u] < size[v] **then**
    clusterId[u]= v; size[v]+ =size[u]
  else
    clusterId[v]= u; size[u]+ =size[v]
  end if
end for
return *clusterId*;

IV. EXPERIMENTS

In this section, we evaluate the CSGCN method. This includes the introduction of datasets and evaluation standards, the experimental implementation details, the comparison methods and the experimental results.

A. EXPERIMENTAL SETTINGS

1) DATASETS

We perform clustering tests on the famous face dataset MS-Celeb-1M [16], which is made up of 100,000 identities and approximately 10 million images. We use a subset of them, which is divided into a training set and a testing set similar to [8], [10] settings. The training set has 86,000 identities and 580,000 face images, and the testing set has a similar size; the testing set and the training set have no overlapping identities.

YouTube-Faces [25] is another commonly used face dataset. There are 3,425 videos in the dataset with a total of 1,595 identities. We use 14,653 face images of 159 identities for training, leaving 1,436 identities with 140,629 face pictures for testing.

We also test our model CSGCN on a relatively large subset of Deepfashion [26]. The training set contains 25,752 pictures of 3,997 categories, and the testing set contains 26,960 pictures of 3,984 categories. There is no overlapping category between the testing set and the training set.

2) EVALUATION METRICS

To assess the performance of the clustering algorithm proposed in this article, we make use of two commonly used evaluation indicators, normalized mutual information (NMI) and Bcubed F-score [47].

As a method of evaluating clustering algorithms, mutual information (MI) is used to measure the similarity between the real label and the label predicted by the model, and normalized mutual information (NMI) adjusts the value of MI between 0 and 1. Therefore, given the real category set T and the predicted category set P, NMI is defined as:

$$NMI(T, P) = \frac{I(T, P)}{\sqrt{H(T)H(P)}}$$

where $H()$ represents the entropy function, and $I(T, P)$ represents the mutual information between set T and set P.

The Bcubed F-score is another common evaluation metric for clustering tasks. Assume that $T(i)$ is the ground-truth label of node i, and $P(i)$ is the label we predicted. $Correct(i, j)$ is defined as the correctness of the pair as:

$$Correct(i, j) = \begin{cases} 1, & if\ T(i) = T(j) \quad and \quad P(i) = P(j) \\ 0, & otherwise \end{cases}$$

The precision rate P is defined as:

$$P = E_i[E_{j:P(j)=P(i)}[Correct(i, j)]],$$

and the recall rate R is defined as:

$$R = E_i[E_{j:T(j)=T(i)}[Correct(i, j)]].$$

Finally, Bcubed F-score is defined as:

$$F = \frac{2PR}{P + R}.$$

3) IMPLEMENTATION DETAILS

In the experiment, we empirically set the K values in MS-Celeb-1M, YouTube-Faces and DeepFashion to 80, 120, and 10, respectively. We use those K values to construct the KNN graph. Since there are too many nodes constructed by the MS-Celeb-1M and YouTube-Faces datasets, our GCN module uses only one hidden layer. In the DeepFashion dataset, the GCN was designed as two hidden layers. We use momentum stochastic gradient descent (SGD), to set the initial learning rate to 0.1, and then the weight decays to $1e^{-5}$.

B. METHOD COMPARISON

We conduct comparative experiments to compare the proposed method with other clustering methods. Considering that we designed the supervised and unsupervised versions, we evaluate them separately. In order to adapt to various methods, in the MS-Celeb-1M dataset, we randomly select one-tenth of the data for testing, which contains 580K images of 8,573 identities. In the YouTube-Faces dataset, we use 1,436 identities with 140,629 face data for testing. We also test a subset of the DeepFashion dataset, which is a challenging long-tailed dataset. For all the methods in the experiment, we adjust various the hyperparameters to obtain the best results.

1) SUPERVISED FACE CLUSTERING

Since most of the existing GCN-based methods are supervised, we first experiment with supervised face clustering. The baselines contain some of the most recent supervised methods as follows:

CDP (consensus-driven propagation) [9] performed clustering by exploiting a more robust pairwise relationship by gathering different predictions.

L-GCN [7] constructed an instance pivot subgraph (IPS) and used GCN for inference to predict the link relationship between two unlabeled samples.

GCN-D [8] is a supervised approach that divides the face clustering problem into GCN-based detection and segmentation modules.

GCN-VE [10] applied GCN to predict the node confidence and edge connectivity, and obtained clustering by connecting each node to the neighbor with the highest connectivity in the candidate set.

CSGCN-S is a GCN-based model based on supervised information proposed in this paper.

The performance of the experimental results in the MS-Celeb-1M, DeepFashion and YouTube-Faces datasets are shown in Tables 1, 2 and 3, respectively. We report the F-score and NMI and report the particular precision rate P and recall rate R for computing the F-score. We also report the running time of the algorithm and the number of clusters formed by clustering in the MS-Celeb-1M and DeepFashion datasets. This further helps us evaluate the strengths and weaknesses of various algorithms. All the methods achieve notable F-score and NMI because they use supervised information and are based on learning. It is worth noting that CDP the only method without GCN. The other methods with GCN, including L-GCN, GCN-D and GCN-VE, are consistently superior to CDP but they are an order of magnitude slower than CDP because of their numerous overlapping subgraphs. Compared with these methods, our CSGCN-S has the best performance and can achieve a computational efficiency close to CDP, which is not based on GCN. Additionally, we discover that the number of clusters formed by L-GCN in MS-Celeb-1M are numerous, while the number of clusters obtained by our method is comparatively conservative. The results of these experiments show that our CSGCN can achieve more accurate clustering results and significantly improve the efficiency of the GCN-based face clustering algorithm.

In addition, we perform statistical analysis on the experimental results of CDP, L-GCN, GCN-D, GCN-VE and CSGCN-S on three datasets to verify whether our method is significantly better than the other supervised algorithms. The Friedman test [48]–[50] is a commonly non-parametric statistical test used to detect differences in treatments across multiple test attempts. Using the Friedman test, we calculate the $p-value = 0.013 < 0.05$, which proves that there are significant differences among the results of these supervised algorithms (0.05 is the significance threshold in statistic, and

TABLE 1. Comparison of our supervised method and other supervised methods on the MS-Celeb-1M dataset. The best results are marked as bold.

Method	P	R	F-score	NMI	Clusters	Time
CDP	82.34	75.37	78.7	94.69	58602	**2.3m**
L-GCN	84.32	84.41	84.37	96.12	44587	86.8m
GCN-D	94.62	78.02	85.52	96.27	74285	62.2m
GCN-VE	95.26	78.53	86.09	96.44	57041	11.5m
CSGCN-S	93.83	81.12	**87.01**	**96.92**	42214	3.8 m

TABLE 2. Comparison of our supervised method and other supervised methods on the DeepFashion dataset. The best results are marked as bold.

Method	P	R	F-score	NMI	Clusters	Time
CDP	72.31	48.18	57.83	90.93	6622	**1.3s**
L-GCN	74.53	50.39	60.13	90.67	9882	23.3s
GCN-D	76.05	48.34	59.11	89.48	9246	13.1s
GCN-VE	78.67	48.57	60.06	90.5	6079	18.5s
CSGCN-S	75.13	50.42	**60.34**	**91.42**	7907	2.1s

TABLE 3. Comparison of our supervised method and other unsupervised methods on the YouTube-Faces dataset. The best results are marked as bold.

Method	P	R	F-score	NMI
CDP	97.14	83.03	89.53	97.59
L-GCN	97.39	84.34	90.39	97.64
GCN-D	96.91	86.36	91.33	97.97
GCN-VE	98.11	84.59	90.84	97.83
CSGCN-S	97.86	85.87	**91.47**	**97.99**

less than 0.05 means that there is a significant difference). Therefore, considering the performance in Tables 1, 2 and 3, our approach is superior to the others.

Next, we further compare the significant differences between our method and each other algorithm in pairs. Generally, there are two commonly used pairwise significance test methods: Nemenyi test [51] and Wilcoxon signed rank test [52]. However, according to the research of [53], when there are five methods for the Nemenyi test, it is necessary to test at least 38 datasets to obtain a meaningful conclusion. In reality, it is very difficult to obtain so many large-scale face datasets at the same time, so this is impractical for our task. Therefore, we use the Wilcoxon signed rank test to compare the differences between the algorithms in pairs. We calculate the pairwise $p-value$ between the two methods and obtain the visualization of $p-value$ matrix below.

As shown in Fig 3, we use visualizations to show the $p-value$ between each supervised methods. As illustrated, the darker the block, the smaller the $p-value$, and the greater the significant difference. The darkest part indicates that the $p-value$ between the two methods is less than 0.05. Apparently, our method is significantly better than other methods. Therefore, we conclude that CSGCN-S has a statistically significant advantage in supervised methods.

2) UNSUPERVISED FACE CLUSTERING

In addition, we extended supervised face clustering to an unsupervised version. The unsupervised baseline is described briefly below.

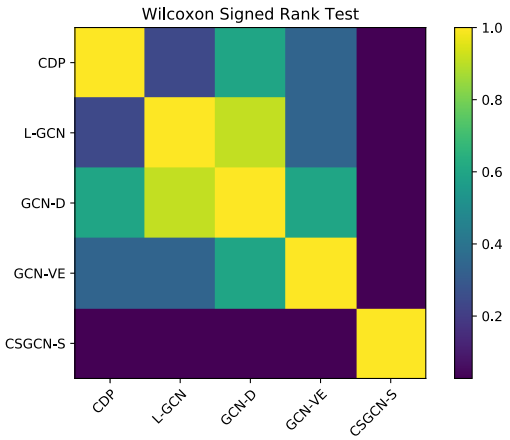


FIGURE 3. The $p-value$ of Wilcoxon signed rank test between the supervised methods in pairs.

K-means clustering [21] is a commonly used classical clustering method. It is an iterative algorithm for clustering analysis.

HAC [23] combined closed clusters in a bottom-up manner based on certain criteria. [24] calculated the distance between two clusters by evaluating the distance between k observations.

DBSCAN [22] selects the clusters based on the density criteria they proposed and takes the sparse outliers as the noise.

ARO (approximate rank-order clustering algorithm) [3] uses an approximate nearest neighbor search and an improved distance metric for clustering.

Spectral clustering [27] uses the eigenvalues and vectors of the graph Laplacian matrix to find clusters.

MeanShift [54] is an iterative process. In short, it is necessary to find data points that belong to the same cluster along the direction of increasing density.

CSGCN-U represents the GCN-based model with the unsupervised information proposed in this paper.

The results are presented in Table 4, 5 and 6. We compare the performances of different unsupervised methods on the MS-Celeb-1M, DeepFashion and YouTube-Faces datasets.

TABLE 4. Comparison of our unsupervised method and other unsupervised methods on the MS-Celeb-1M dataset. The best results are marked as bold.

Method	P	R	F-score	NMI	Clusters	Time
K-means	81.33	79.23	80.26	94.54	8573	11.5h
HAC	98.29	54.91	70.46	92.92	122754	12.7h
DBSCAN	99.03	50.81	67.17	92.31	134744	**1.9m**
ARO	99.69	9.12	16.71	84.64	287267	27.5m
CSGCN-U	93.55	81.21	**86.94**	**96.69**	44427	3.8 m

TABLE 5. Comparison of our unsupervised method and other unsupervised methods on the DeepFashion dataset. The best results are marked as bold.

Method	P	R	F-score	NMI	Clusters	Time
K-means	55.54	51.93	53.68	89.02	3991	936s
HAC	92.28	33.14	48.77	90.44	17410	110s
DBSCAN	87.66	38.22	53.23	90.75	14350	2.2s
ARO	67.21	43.66	52.94	88.63	10376	7.5s
MeanShift	72.68	46.52	56.73	89.29	8435	2.2h
Spectral	75.81	33.43	46.4	86.95	2504	2.1h
CSGCN-U	76.2	49.26	**59.84**	**91.16**	8282	**2.1s**

TABLE 6. Comparison of our unsupervised method and other unsupervised methods on the YouTube-Faces dataset. The best results are marked as bold.

Method	P	R	F-score	NMI
K-means	86.40	67.83	76.00	94.03
HAC	99.69	78.73	87.98	97.19
DBSCAN	98.31	80.78	88.69	97.41
ARO	99.84	60.6	75.42	94.35
CSGCN-U	98.16	84.03	**90.54**	**97.69**

For K-means, although it has achieved good results, it is necessary to specify the number of clusters in advance, and the number has a great impact on the results, which means that it is difficult to use in a reality where the number of clusters cannot be clearly defined. HAC has a lower F-score and forms too many clusters, and it is also the most time-consuming method in MS-Celeb-1M. DBSCAN is efficient but because it assumes that clusters should have similar densities, it generates too many clusters. ARO depends on the number of neighbors. It forms the most number of clusters in MS-Celeb-1M. In addition, MeanShift and Spectral clustering perform well on DeepFashion but they take a very long time to converge, thus limiting their application. Therefore, we do not use them for the MS-Celeb-1M and YouTube-Faces datasets. Obviously, our CSGCN-U method is significantly better than all the other conventional unsupervised clustering methods. Notably, our model does not need to confirm the number of clusters in advance. Owing to the use of an improved GCN for feature embedding, our unsupervised model achieves very good results.

Moreover, we also perform statistical analysis on the experimental results of K-means, HAC, DBSCAN, ARO and CSGCN-U to verify whether our method is significantly different from other unsupervised algorithms. The Friedman test is used to calculate that the $p-value = 0.002 < 0.05$, which proves that there are significant differences among these algorithms. Therefore, our approach is superior to the others considering that it has the best F-score and NMI on the three datasets. To further evaluate the pairwise significant difference between algorithms, we employ the Wilcoxon signed rank test again to calculate the $p-value$ between the algorithms in pairs.

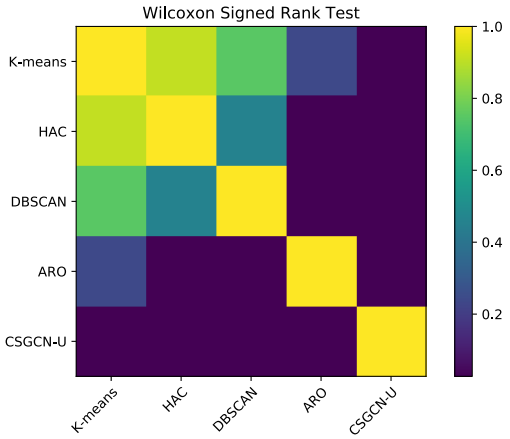


FIGURE 4. The $p-value$ of Wilcoxon signed rank test between the unsupervised methods in pairs.

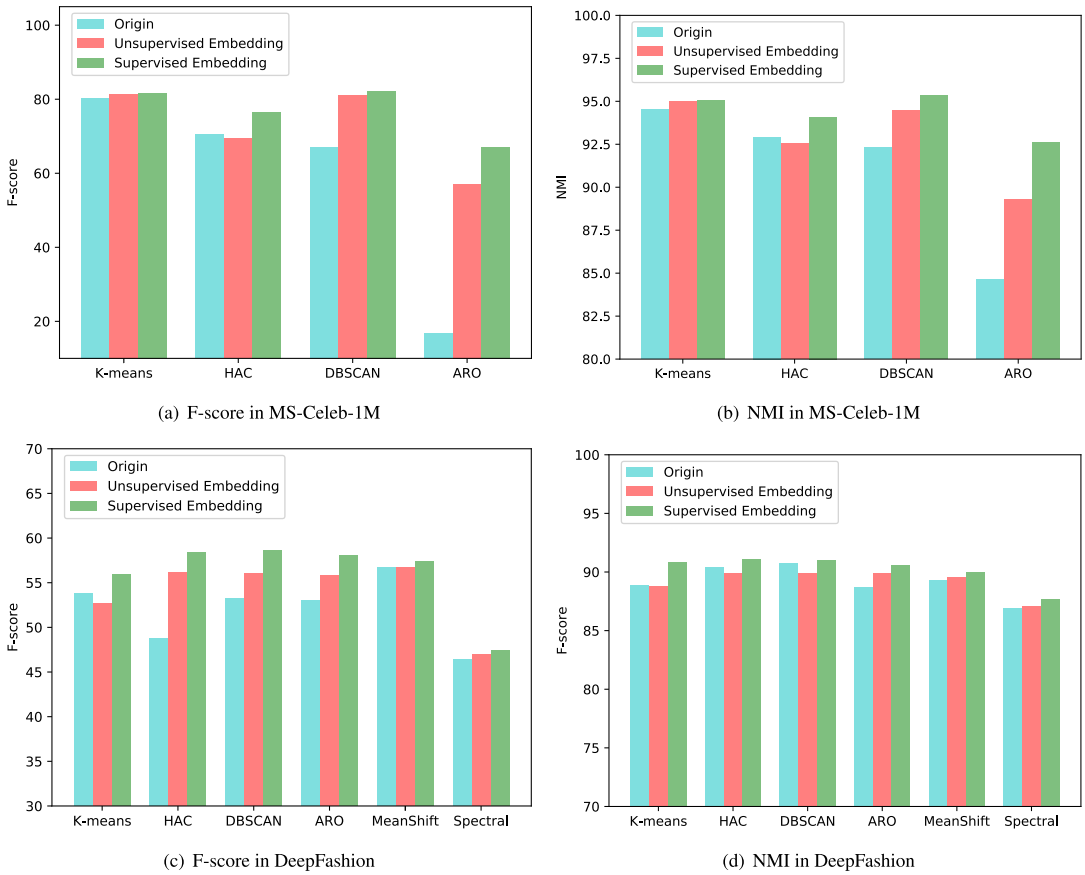
As shown in Fig 4, we use visualizations to show the $p-value$ between the two methods. It can be seen that our method is significantly better than the other methods intuitively. Therefore, we conclude that CSGCN-U has a statistically significant advantage in unsupervised methods.

C. ANALYSIS FOR EMBEDDING FEATURES

In order to further verify the validity of our embedded features, we study feature discriminative power and feature distribution on two datasets.

1) FEATURE DISCRIMINATIVE POWER

First, we conduct experiments on the MS-Celeb-1M and DeepFashion datasets to explore the feature discriminative power. As shown in Fig 5, for the feature embedding obtained, we test the efficiency gain of the obtained supervised and unsupervised embedding on the traditional



(a) F-score in MS-Celeb-1M

(b) NMI in MS-Celeb-1M

(c) F-score in DeepFashion

(d) NMI in DeepFashion

FIGURE 5. Embedding feature discriminative power on MS-Celeb-1M (the top row) and DeepFashion (the bottom row).

clustering method. Specifically, we save the supervised and unsupervised feature embeddings obtained in the inference process, and use the traditional clustering methods later.

The accuracy is significantly improved, which shows that GCN is used to learn the local information between nodes, making the features more in line with the clustering standard. It can be seen from the figure that ARO has the most obvious improvement effect. Our feature embedding can directly improve the F-score and NMI performances of ARO to close to the average level.

Overall, the performance gains shown by the supervision model are more obvious.

2) FEATURE DISTRIBUTION ANALYSIS

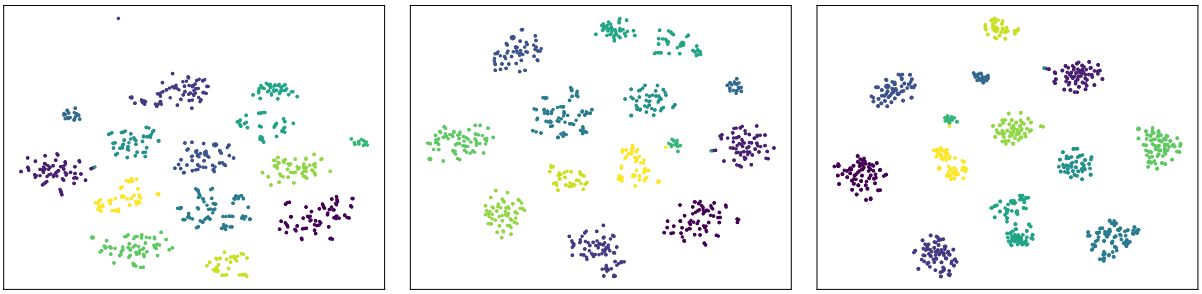
In addition, we conduct experiments to analyze the embedded feature distributions. Considering that this paper focuses on face clustering, we only conduct this experiment on the face dataset, MS1M. As shown in Fig 6, to observe the data distribution in the embedding space, we select 12 identities and put their features into the t-SNE to visualize their distribution. (a) shows the original features (after CNN extracted features), (b) shows the graph convolution module using unsupervised loss to the embedding feature, and (c) uses supervised loss.

From (a), we can see that these samples are scattered together, and it is difficult to determine the division of categories without color labeling. After the GCN embeds the featurses of the data, the features are more compact. Similar faces gather together more, whereas dissimilar faces will be scattered. Therefore, the clustering results obtained are more accurate. It can also be seen from the figure that the distance between clusters after the supervised embedding feature visualization is smaller, which is more conducive to clustering.

D. ABLATION STUDY

The CSGCN we proposed includes two main influencing factors, the confidence score and graph convolution. To verify the effectiveness of each component, we carefully perform two kinds of ablation experiments and compare their performance. First, we explore different values of λ in the confidence design described in Sec.III.B, as shown in Table 7, when $\lambda = 1$, we obtain a higher precision but a lower recall, while when $\lambda = 0$, we get a lower precision value but a higher recall, and when $\lambda = 0.5$, we obtain equilibrium and the highest F-score and NMI.

Furthermore, graph convolution is another important component of the proposed CSGCN. Therefore, we also compare the simple graph convolution module with the traditional GCN in terms of both accuracy and efficiency, and the results are shown in Table 8. Here, CSGCN-NL means that the traditional graph convolution module is used, that is, the nonlinearity in GCN is retained. Instead, CSGCN uses the sample graph convolution module, which eliminates the nonlinearity



(a) Original (b) Unsupervised Embedding (c) Supervised Embedding

FIGURE 6. Feature distribution visualization on t-SNE. Several identities are illustrated. Nodes of the same color indicates faces of the same identity.

TABLE 7. Comparison on MS-Celeb-1M with different value of λ in the confidence.

Method	P	R	F-score	NMI
CSGCN($\lambda = 1$)	95.24	78.1	85.82	96.4
CSGCN($\lambda = 0.75$)	94.64	79.06	86.15	96.54
CSGCN($\lambda = 0.5$)	93.83	81.12	87.01	96.92
CSGCN($\lambda = 0.25$)	94.24	80.11	86.68	96.47
CSGCN($\lambda = 0.0$)	93.86	79.42	86.04	96.46

TABLE 8. The effect of removing the non-linear layer in GCN on DeepFashion.

Method	P	R	F-score	NMI	Test-Time	Train-Time
CSGCN-NL	75.24	50.38	60.35	91.44	2.6s	12h
CSGCN	75.13	50.42	60.34	91.42	2.1s	7h

of the network. After removing the nonlinear layer of the GCN, the accuracy of the CSGCN decreases slightly, but the efficiency of the model is greatly improved. Therefore, using the CSGCN can significantly reduce training time with only a slight impact on accuracy.

V. CONCLUSION

In this paper, a new face clustering framework is proposed, which greatly optimizes the heuristic step problems and addresses the large number of overlapping subgraphs in previous methods. The proposed method consists of two modules; the graph convolution module predicts the confidence of the nodes and embedding features, and the clustering module is responsible for clustering the data based on the results obtained by the previous module. Our method can complete clustering in an unsupervised way without data annotation. The proposed method significantly improves the accuracy and efficiency of face clustering. In addition, experiments on the Deepfashion dataset show that our method has application prospects in datasets other than facial datasets.

In the future, we will conduct further experimental explorations on the hyperparameter settings of the model. We will also try to further expand the framework of the model to excavate more useful information. In addition, due to the influence of unfavorable external factors, there will be considerable noise in the face photos. Therefore, in the future, we need to focus on how to reduce the impact of noise on the model, and improve its robustness.

REFERENCES

[1] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, "The devil of face recognition is in the noise," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 765–780.

[2] W.-A. Lin, J.-C. Chen, C. D. Castillo, and R. Chellappa, "Deep density clustering of unconstrained faces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8128–8137.

[3] C. Otto, D. Wang, and A. K. Jain, "Clustering millions of faces by identity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 289–303, Feb. 2017.

[4] Y. Shi, C. Otto, and A. K. Jain, "Face clustering: Representation and pairwise constraints," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1626–1640, Jul. 2018.

[5] C. Zhu, F. Wen, and J. Sun, "A rank-order distance based clustering algorithm for face tagging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 481–488.

[6] Y. He, K. Cao, C. Li, and C. Loy, "Merge or not? Learning to group faces via imitation learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 6902–6909.

[7] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1117–1125.

[8] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to cluster faces on an affinity graph," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2298–2306.

[9] X. Zhan, Z. Liu, J. Yan, D. Lin, and C. C. Loy, "Consensus-driven propagation in massive unlabeled data for face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 568–583.

[10] L. Yang, D. Chen, X. Zhan, R. Zhao, C. C. Loy, and D. Lin, "Learning to cluster faces via confidence and connectivity estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13369–13378.

[11] S. Guo, J. Xu, D. Chen, C. Zhang, X. Wang, and R. Zhao, "Density-aware feature embedding for face clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6698–6706.

[12] C. Qi, J. Zhang, H. Jia, Q. Mao, L. Wang, and H. Song, "Deep face clustering using residual graph convolutional network," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106561.

[13] S. Shen, W. Li, Z. Zhu, G. Huang, D. Du, J. Lu, and J. Zhou, "Structure-aware face clustering on a large-scale graph with 107 nodes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9085–9094.

[14] J. Ye, X. Peng, B. Sun, K. Wang, X. Sun, H. Li, and H. Wu, "Learning to cluster faces via transformer," 2021, *arXiv:2104.11502*.

[15] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang, "EasyAlbum: An interactive photo annotation system based on face clustering and re-ranking," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2007, pp. 367–376.

[16] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 87–102.

[17] A. Nech and I. Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7044–7053.
[18] Y. Zhang, W. Deng, M. Wang, J. Hu, X. Li, D. Zhao, and D. Wen, "Global-local GCN: Large-scale label noise cleansing for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7731–7740.
[19] B. Debnath, G. Coviello, Y. Yang, and S. Chakradhar, "UAC: An uncertainty-aware face clustering algorithm," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 3487–3495.
[20] J. C. Klontz and A. K. Jain, "A case study of automated face recognition: The Boston marathon bombings suspects," *IEEE Comput.*, vol. 46, no. 11, pp. 91–94, Nov. 2013.
[21] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Knowl. Discovery Data Mining (SIGKDD)*, 1996, vol. 96, no. 34, pp. 226–231.
[23] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, Jan. 1973.
[24] P. Yildirim and D. Birant, "*K*-Linkage: A new agglomerative approach for hierarchical clustering," *Adv. Electr. Comput. Eng.*, vol. 17, no. 4, pp. 77–88, 2017.
[25] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 529–534.
[26] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1096–1104.
[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[28] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2013, pp. 160–172.
[29] H. Averbuch-Elor, N. Bar, and D. Cohen-Or, "Border-peeling clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1791–1797, Jul. 2020.
[30] M. Du, R. Wang, R. Ji, X. Wang, and Y. Dong, "ROBP a robust border-peeling clustering using Cauchy kernel," *Inf. Sci.*, vol. 571, pp. 375–400, Sep. 2021.
[31] S. A. Shah and V. Koltun, "Robust continuous clustering," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 37, pp. 9814–9819, 2017.
[32] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2003, p. I.
[33] W.-A. Lin, J.-C. Chen, and R. Chellappa, "A proximity-aware hierarchical clustering of faces," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 294–301.
[34] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.
[35] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.
[36] L. Backstrom and J. Leskovec, "Supervised random walks: Predicting and recommending links in social networks," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 635–644.
[37] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5115–5124.
[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[39] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
[40] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
[41] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7444–7452.
[42] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin, "Convolutional sequence generation for skeleton-based action synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4394–4402.
[43] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," 2018, *arXiv:1801.10247*.
[44] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6861–6871.
[45] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, 1999.
[46] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[47] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Inf. Retr.*, vol. 12, no. 4, pp. 461–486, 2009.
[48] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
[49] M. Friedman, "A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 34, no. 205, p. 109, Mar. 1939.
[50] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, Mar. 1940.
[51] P. B. Nemenyi, *Distribution-Free Multiple Comparisons*. Princeton, NJ, USA: Princeton Univ. Press, 1963.
[52] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. Berlin, Germany: Springer, 1992, pp. 196–202.
[53] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[54] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

DENGDI SUN received the B.Eng. degree in statistics, the M.Eng. degree in mathematics, and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2005, 2008, and 2012, respectively. He was a Visiting Scholar with the Department of Computing Science and Mathematics, University of Stirling, Scotland, U.K., in 2013, and the Department of Computer Science and Engineering, University of Texas at Arlington, TX, USA, in 2017. Currently, he is an Associate Professor with the School of Computer Science and Technology, Anhui University. His research interests include computer vision, machine learning, and deep learning.

KANG YANG received the bachelor's degree from the School of Mathematical Science, Huaibei Normal University, Huaibei, China, in 2018. He is currently pursuing the master's degree with the School of Computer Science and Technology, Anhui University. His current research interests include graph learning and computer vision.

ZHUANLIAN DING received the M.S. and Ph.D. degrees from the School of Computer Science and Technology, Anhui University, Hefei, China, in 2014 and 2018, respectively. She is currently a Lecturer with the School of Internet, Anhui University. Her current research interests include computer vision, pattern recognition, and graph learning.