Knowledge-Based Systems 211 (2021) 106561

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Deep face clustering using residual graph convolutional network

Chao Qi [a], Jianming Zhang [a], Hongjie Jia [a,b,*], Qirong Mao [a,b], Liangjun Wang [a], Heping Song [a]

[a] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
[b] Jiangsu Engineering Research Center of big data ubiquitous perception and intelligent agriculture applications, Zhenjiang 212013, China

ARTICLE INFO

Article history:
Received 23 May 2020
Received in revised form 25 October 2020
Accepted 26 October 2020
Available online 27 October 2020

Keywords:
Face clustering
kNN
Deep GCNs
Residual network

ABSTRACT

Face clustering has important applications in image retrieval and criminal investigation. Face images can be seen as the nodes of a graph and the possibility of links between the nodes will help us find clusters. Graph Convolutional Networks (GCNs) are powerful tools to infer the possibility of linkage between a given node and its neighbors. However, existing face clustering methods use shallow GCNs and have limited learning capabilities. We propose a deep face clustering method using Residual Graph Convolutional Network (RGCN), which contains more hidden layers. For each node, *k*-Nearest Neighbor (*k*NN) algorithm is used to construct its sub-graphs. Then we apply the idea of ResNet into GCNs and construct RGCN to learn the possibility of linkage between two nodes. Compared with other popular face clustering approaches, our method is more efficient and has better clustering results in the experiments. In addition, the proposed RGCN clustering approach is able to detect the quantity of clusters automatically and can be extended to large datasets.

1. Introduction

Face recognition has been remarkably developed and the accuracy of it has reached a high level in recent years. However, it is worth noting that the modern advanced face recognition technology largely depends on the large-scale training datasets with ground truth labels. A large number of face images can be easily collected from the Internet, but the annotation of face images is very time-consuming. Therefore, developing unlabeled data through unsupervised learning becomes an attractive choice and has aroused great interest in academia and industry [1,2].

Clustering is a fundamental and effective tool for processing large-scale unlabeled data. Traditional clustering algorithms, e.g. *K*-means [3], Spectral Clustering [4] and DBSCAN [5], assign unlabeled datasets to "pseudo-labels" so that they can be used as labeled data. However, the drawbacks of these algorithms limit their application in real world face clustering problems. For example, Spectral Clustering has no requirements for data distribution but is sensitive to parameter selection, *K*-means requires that the dataset is the convex dataset, and DBSCAN requires that the distribution of data is relatively uniform and the density is relatively close.

In contrast, many connection-predict clustering algorithms have no requirements on data structures and can get better clustering results [6]. Connection-predict approaches will judge if two points have connection or not. Wang et al. [7] point out that the connection possibility between a point and its neighbors can be extracted from the context. GCNs is able to learn important information from the context of graph. Inspired by the success of deep CNN models, we try to design a deep GCN model for face clustering. However, stacking more layers into GCNs will encounter the vanishing gradient and network degradation problem, finally the features of graph nodes converges to the same value [8]. As the number of network layers increases, the loss of training set gradually decreases and then tends to saturate. If the depth of network is further increased, loss of training set will increase instead. From the perspective of information theory, due to the existence of Data Processing Inequality (DPI), the image information contained in the feature map decreases layer by layer as the number of layers is deepened during forward transmission. Therefore, current GCN models are usually less than four layers [9].

We find that when the deep CNNs is degraded, the shallow network can achieve better training performance than the deep network. At this time, if we transmit the features from the lower layers to the higher layers, the effect should be at least not worse than the shallow network, which is the idea of ResNet. ResNet [10] ease the vanishing gradient problem in deep CNNs by adding residual connections between input and output layers. Inspired by ResNet, we propose a residual graph convolutional network RGCN to solve the gradient vanishing problem in deep GCNs. RGCN adds direct mapping between different layers of the

* Corresponding author at: School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China.
E-mail address: jiahj@ujs.edu.cn (H. Jia).

https://doi.org/10.1016/j.knosys.2020.106561
0950-7051/

deep GCNs to ensure that the $l+1$ layer network contains more image information than the l layer.

Then we develop a deep face clustering framework using RGCN, in which residual connections are used for training very deep GCNs. We summarize the framework as follows. First of all, we transform the clustering task into the task of judging the link probability between nodes. Secondly, we construct sub-graph for each node with their nearest neighbors. The sub-graph includes one-hop neighbors and two-hop neighbors, and we may use brute force search or KDTree method to find them. Then we use RGCN to learn the graph context information of each node and infer whether the two nodes are linked. Finally, RGCN outputs the connection possibility of the central node with its nearest neighbors, and we merge the nodes having positive links to get a cluster. The effect of RGCN to the accuracy and stability of the whole deep face clustering framework is analyzed extensively. Experiments show that the proposed method has better clustering results on large-scale face datasets compared with the latest face clustering approaches.

The main contributions are in three aspects: (1) We propose a residual graph convolutional network RGCN, which avoids the vanishing gradient and network degradation problem when training deep GCN model. RGCN can make full use of the structural information in the graph for clustering. (2) We construct a deep face clustering framework based on RGCN, in which RGCN is used to infer the connection possibility among graph vertices. To the best of our knowledge, it is the first time to apply deep GCNs into face clustering. (3) We test the effectiveness of the proposed deep face clustering framework on benchmark face datasets and our method achieves state-of-the-art performance compared with the latest face clustering approaches.

This paper is organized as follows: Section 2 introduces the related works about face clustering and GCNs; Section 3 shows the details of the proposed RGCN face clustering method; Section 4 analyzes the experimental results of different clustering algorithms; Section 5 is conclusion and future work.

2. Related work

2.1. Face clustering

Clustering is a common task in unsupervised learning. Some conventional and common clustering methods like K-means, spectral clustering and DBSCAN, are unsuitable to face clustering task because of the complexity of face feature distribution. Agglomerative Hierarchical Clustering (AHC) [11–13] has made great breakthroughs on face clustering task in recent years. Lin et al. [12] propose a hierarchical clustering method using a linear support vector machine to classify local negative and positive samples, which are used to group face images. A density-aware cluster-level affinity measure also designed by Lin et al. [13] applies singular value decomposition to process density-unbalanced data. Zhu et al. [14] turn the face clustering problem into a multiple graph cut problem and use the gradient flow approach to optimize the objective function. Tapaswi et al. [15] propose a ball cluster learning algorithm for face clustering, which can automatically determine the number of clusters in videos. All the above methods have limitations on large-scale clustering task due to their high computation complexity.

In recent works, the Approximate Rank-Order clustering algorithm (ARO) [1] successfully lifts the restrictions of large-scale clustering. The key idea of ARO is to calculate the distance of only the first k nearest faces. Compared with other clustering algorithms, ARO is much more efficient because its computational complexity is only $O(kn)$. The face clustering based on GCNs are also applied to process large-scale face datasets. Yang et al. [16] propose a clustering framework, in which an affinity graph is built for GCNs to detect face cluster. Wang et al. [7] propose a Linkage-Based Face clustering algorithm (LBF), which exploits GCNs to infer the likelihood of linkage between two nodes. This is a new breakthrough in GCNs-based face clustering. The clustering results of LBF are much better than other clustering methods because GCNs has an amazing classification ability in determining whether two nodes belong to the same cluster. However, the learning ability of LBF algorithm is limited due to the shallow GCNs structure used in the clustering framework. In this paper, we try to improve the performance of face clustering by deepening the layers of GCNs.

2.2. Graph convolutional network

In many machine learning problems, the input is graph-structured data. GCNs have better performance on graph-structured data compared with CNNs. There are two ways to understand GCNs: spectral methods [17–19] and spatial methods [20,21]. The core idea of a spatial methods is to aggregate the information of neighbor nodes, while spectral based GCNs mainly uses graph Fourier transform to generalize convolution. The existing work shows that GCNs can greatly improve the performance of various tasks. For instance, the GCNs using spectral graph convolution proposed by Kipf et al. [19] obtain amazing results on semi-supervised classification tasks. Hamilton et al. [22] show that GCNs have better performance in feature representations compared with other methods. Liu et al. [23] successful apply GCNs in link prediction.

However, current GCN algorithms are limited to shallow depths. Recent works attempt to train deeper GCNs. Li et al. [9] study the limitations of deep GCNs and show that more layers in the network may cause vanishing gradient and network degradation problem. To solve the problem in deep GCNs, we propose RGCN for face clustering according to the idea of ResNet. RGCN has deeper networks and alleviates the vanishing gradient problem by adding residual connections between inputs and outputs of layers. In this paper, we design a deep face clustering framework, which adopts RGCN to process the graph constructed by kNN and predict the possibility of linkage between face nodes.

3. Methodology

3.1. Framework overview

The features of face images are extracted by CNN and we have the feature space of face images $X = [x_1, x_2, \ldots, x_N]^{\mathrm{T}} \in \mathrm{R}^{N\times D}$, where D represents the number of features, N is the number of images. We construct sub-graph for each node by finding its $k1$ direct neighbors and $k2$ single indirect neighbors. The possibility of linkage between two nodes is inferred through RGCN.

The proposed deep face clustering framework is shown in Fig. 1. Next we will introduce each part of the proposed framework respectively. The sub-graph construction for each node is described in Section 3.2. Given the sub-graphs as input data, we adopt RGCN to infer the possibility of linkage between a given node and its neighbors. The mechanism of RGCN is presented in Section 3.3. Finally, the linked nodes are merged into the clusters transitively according to a set of weighted edges output by RGCN and the details are discussed in Section 3.4.

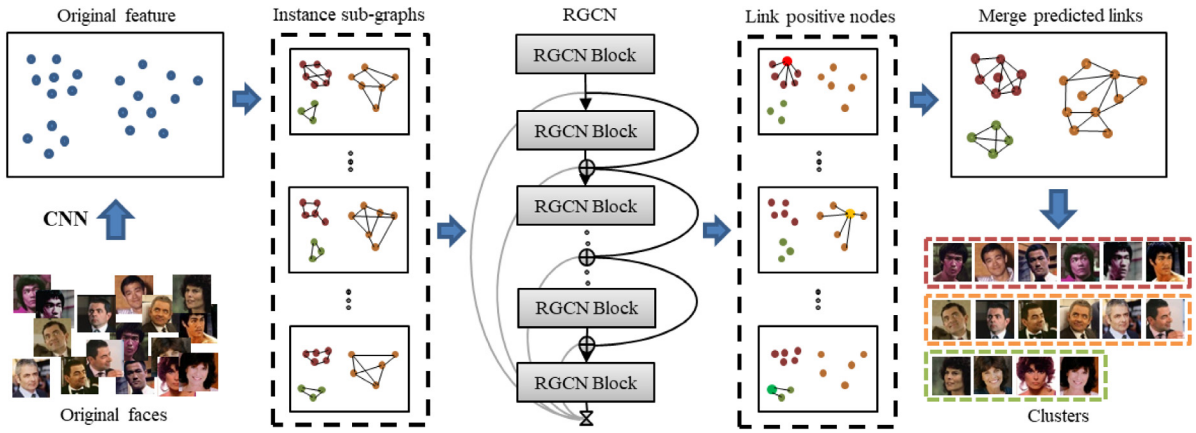

Fig. 1. Deep face clustering framework based on RGCN.

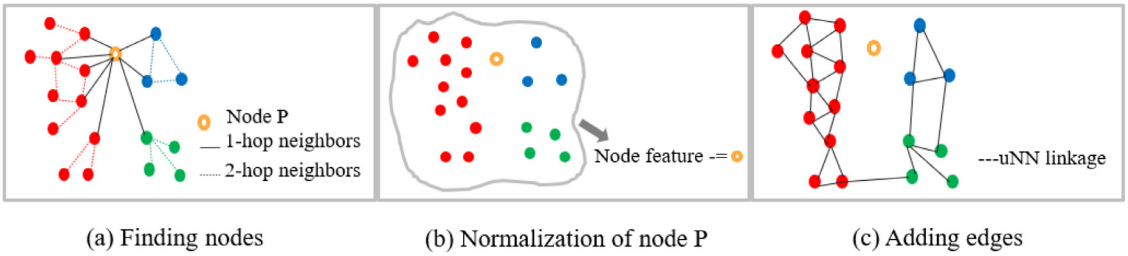

Fig. 2. Framework for constructing sub-graphs. (a) Given a node P, we find its direct neighbors and single indirect neighbors. (b) The features of these nodes are normalized by subtracting the feature of node P. (c) We find uNNs of node P from the entire feature space.

3.2. Construction of sub-graph

In this paper, we construct a sub-graph for each node by finding its k nearest neighbors. The sub-graph construction is generated by three steps, as shown in Fig. 2.

First of all, we can get k nearest neighbors of node P through brute force search or KDTree algorithm. We construct a sub-graph for each node by finding its $k1$ direct neighbors and $k2$ single indirect neighbors. Since the h-indirect neighbors contain very little information when $h \geq 2$, we only select the direct neighbors and single indirect neighbors of each node to construct sub-graphs so as to reduce the complexity of calculation.

Secondly, we have the set V_p composed of the direct neighbors and single indirect neighbors of central node P and their node features $\{F_q | \ q \in V_p\}$. In order to take advantage of the mutual information between nodes, a normalization process is needed. The normalization rule is as follows:

$$X_p = \left[\ldots, F_q - F_p, \ldots\right]^{\mathrm{T}}, \ \text{for all } q \in V_p \tag{1}$$

where F_p is the feature of node P and X_p is used to describe the normalized node features. We use the node features F_q minus the feature of central node P to get normalized node features Xp.

In the last step, we add potentially connected edges among all nodes in V_p. We seek out the top u nearest neighbors for node q belonging to V_p in the entire feature space. An edge (q, r) is added to the edge set E_p if the node r in uNNs appears in V_p. Finally, we complete the construction of the sub-graph of node P. We can get the feature matrix X_p and the adjacency matrix $A_p \in \mathrm{R}^{|V_p| \times |V_p|}$.

3.3. Residual graph convolutional network

The constructed sub-graph of the central node P contains information about the surrounding nodes. To utilize the neighborhood information, we may use GCNs to process the sub-graph, because GCNs can achieve good results in node classification [7]. However, most GCNs are shallow models with limited learning ability. So we considered whether we could improve the network performance by deepening the network like CNNs. Traditional deep GCNs models are difficult for training. For GCNs, increasing the number of graph convolutional layers may cause the vanishing gradient and network degradation problem. To address this issue, we design a deep GCN model called RGCN using residual learning technics. Then we construct our face clustering framework according to RGCN to improve the clustering qualities of face images. The proposed RGCN model consists of a series of RGCN blocks, which add residual connections between input and output layers. The detailed structure of a RGCN block is shown in Fig. 3.

As shown in Fig. 3, the input of a RGCN block is the sub-graphs of nodes. The RGCN block consists of four graph convolution layers, because this structure achieves the best clustering results in our experiments. The expression of a graph convolution layer is:

$$\mathbf{Y} = \varphi\left([\mathbf{X} \parallel \mathbf{FX}]\,\mathbf{W}\right) \tag{2}$$

where $\mathbf{X} \in \mathrm{R}^{N \times din}$ is the feature matrix, $\mathbf{Y} \in \mathrm{R}^{N \times dout}$ is the output matrix, N represents the quantity of nodes, din represents the number of features of input nodes and $dout$ represents the number of features of output nodes. $\mathbf{W}$ is a $2din \times dout$ weight matrix learned for the graph convolution layer. $\varphi(\cdot)$ represents the transfer function, such as ReLU function. $\mathbf{F} = f\ (\mathbf{X}, \mathbf{A})$ is an $N \times N$ matrix calculated by the aggregation function $f(\cdot)$ of $\mathbf{X}$ and $\mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix. The input of Eq. (2) is matrix $\mathbf{X}$ and $\mathbf{A}$. The feature matrix Xp of Eq. (1) is the input of the first graph convolution layer.

The aggregation function $f(\cdot)$ has three categories: mean aggregation, weighted aggregation and attention aggregation. Here we use the mean aggregation method to calculate matrix $\mathbf{F}$ and the formula is as follows:

$$\mathbf{F} = \mathbf{\Lambda}^{-1/2}\mathbf{A}\mathbf{\Lambda}^{-1/2} \tag{3}$$

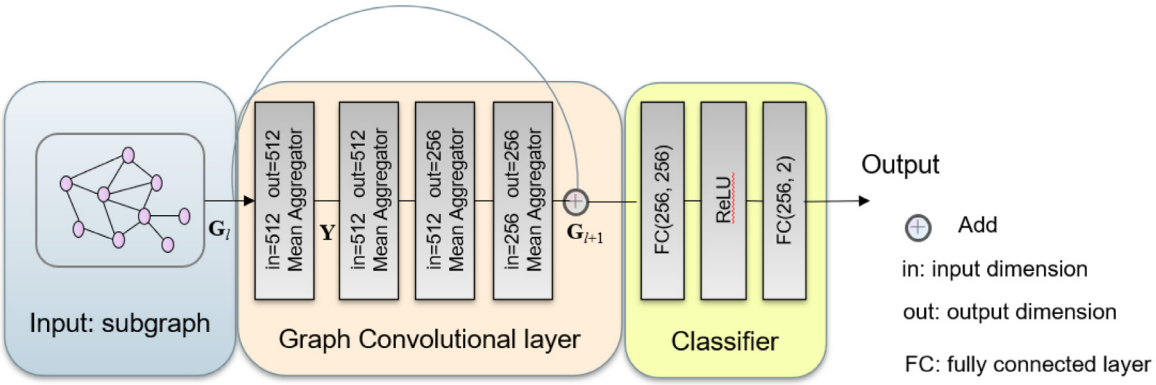

Fig. 3. Structure of the RGCN block.

where Λ represents a diagonal matrix with $\Lambda_{ii} = \sum_j \mathbf{A}_{ij}$ and $\mathbf{A}$ represents the adjacency matrix.

In the RGCN block, we design a shortcut for every four graph convolution layers, so that we can make better use of the graph structure information and input features. A RGCN block can be expressed as:

$$\mathbf{G}_{l+1} = \mathbf{G}_l + \Phi\left(\mathbf{G}_l, \mathbf{W}_l\right) \tag{4}$$

where $\mathbf{G}_l$ is the direct mapping, which is the input of the block; $\mathbf{W}_l$ is the learnable parameter of the layer l; $\Phi(\mathbf{G}_l, \mathbf{W}_l)$ is the residual mapping, which is composed of four convolution operations; $\mathbf{G}_{l+1}$ is the output of the block.

For a deeper layer L, its relationship with layer l can be expressed as:

$$\mathbf{G}_L = \mathbf{G}_l + \sum_{i=l}^{L-1} \Phi(\mathbf{G}_i, \mathbf{W}_i) \tag{5}$$

Eq. (5) reflects two characteristics of the RGCN:

(1) The L-layer can be expressed as the sum of any shallow l-layer and their residual parts;

(2) $\mathbf{G}_L = \mathbf{G}_0 + \sum_{i=0}^{L-1} \Phi(\mathbf{G}_i, \mathbf{W}_i)$, $\mathbf{G}_L$ is the cumulative sum of the individual residual block features, while the Multi-Layer Perceptron (MLP) is the cumulative product of the feature matrix.

According to the derivative chain rule used in back-propagation, the gradient of loss function with respect to $\mathbf{G}_l$ can be expressed as:

$$\frac{\partial \text{loss}}{\partial \mathbf{G}_l} = \frac{\partial \text{loss}}{\partial \mathbf{G}_L}\frac{\partial \mathbf{G}_L}{\partial \mathbf{G}_l} = \frac{\partial \text{loss}}{\partial \mathbf{G}_L}\left(1 + \frac{\partial}{\partial \mathbf{G}_l}\sum_{i=l}^{L-1} \Phi(\mathbf{G}_i, \mathbf{W}_i)\right) \tag{6}$$

where $\frac{\partial \text{loss}}{\partial \mathbf{G}_L}$ represents the gradient of the loss function to the L layer, and the 1 in parentheses indicates a short-circuit mechanism that propagates the gradient losslessly, which indicates that the gradient of the L layer can be directly transferred to any layer l that is shallower than it. $\frac{\partial}{\partial \mathbf{G}_l}\sum_{i=l}^{L-1} \Phi(\mathbf{G}_i, \mathbf{W}_i)$ is the residual gradient. In the training process, the residual gradient is not directly transferred, it needs to pass through the layer with weight, which cannot be $-\mathbf{1}$ all the time. Moreover, even if the residual gradient is relatively small, the presence of $\mathbf{1}$ will not let the gradient to disappear.

By analyzing the two processes of forward and backward propagation of RGCN, we find that when the residual block satisfies the direct mapping assumption, information can propagate between the higher and lower layers very smoothly. This indicates that direct mapping is a sufficient condition that allows RGCN to train deep GCNs, and RGCN can be a good solution to the gradient vanishing problem in deep GCNs.

3.4. Clustering

We use RGCN to process the sub-graph constructed in Section 3.2 and obtain a series of edge weights between nodes, which represent the possibility of the two nodes connecting with each other. To obtain clustering, a simple method is to cut out all edges with weights less than a certain threshold and then use the BFS algorithm to propagate the pseudo-label. However, the most obvious question is whether the threshold is selected properly directly affects the final performance. To solve this problem, the pseudo label propagation strategy proposed in [24] is used: variable thresholds and the maximum number of merges are used to prevent excessive clustering in one class of clustering results. The clustering rule is as follows: the initial threshold is the minimum value of the edge weight. The threshold value become larger as the iteration times increase. In each iteration, we choose the edges whose weights are larger than the threshold. If the cluster size exceeds the maximum merge size, the edge is left undetermined until the next iteration. In fact, our proposed method produces many singleton clusters which contain only a single sample. We check these singleton samples and find that most of them are difficult to identify. For example, these singleton samples may include blurred faces or low-resolution faces. Therefore, in the final clustering process, we filter out all singleton samples to obtain better clustering results.

4. Experiments

4.1. Datasets and evaluation metrics

In this section, different datasets are used for the training of clustering models. We train the CNN model based on ArcFace [25] on the union set of MS-Celeb-1M [26] and VGGFace2 [27] dataset in order to extract face features. A random subset of CASIA dataset [28] containing 200k faces with 5k identities is used for RGCN training. The IJB-B dataset [29] is used for RGCN testing because it contains protocols for clustering tasks. Seven subtasks with different number of ground truth identities make up this protocol. Different subtasks have different numbers of face images and distinct identities. We perform experiments on three subtasks with 512, 1024, and 1845 identities. The detail information of these datasets is given in Table 1.

To evaluate the performance of clustering algorithms, two common measures are used: Normalized Mutual Information (NMI) [30] and BCubed F-measure [31].

(1) NMI. NMI is a frequently-used evaluation index for measuring the normalized similarity between labels assigned by clustering algorithms and ground truth labels. Suppose U_t represents

Table 1
Training and testing datasets used in the experiments.

Dataset	# Samples	# Identities
CASIA	500k	10k
IJB-B-512	18,171	512
IJB-B-1024	36,575	1024
IJB-B-1845	68,195	1845

the true classes of dataset and U_c represents the generated clusters by clustering algorithms. The NMI of U_t and U_c can be calculated by Eq. (7):

$$\mathrm{NMI}(U_t, U_c) = \frac{\mathrm{I}(U_t, U_c)}{\sqrt{\mathrm{H}(U_t)\mathrm{H}(U_c)}} \tag{7}$$

where $\mathrm{H}(\bullet)$ is the entropy function of a data assignment, $\mathrm{I}(U_t, U_c)$ is the mutual information of U_t and U_c.

(2) BCubed F-measure. F-measure is another practical clustering evaluation metric. Assume $U_t(i)$ is the true class label of point i, $U_c(i)$ is the cluster label of point i. Given two points i and j, they have the following relations:

$$\mathrm{Correct}(i,j) = \begin{cases} 1, & \text{if } U_t(i) = U_t(j) \text{ and } U_c(i) = U_c(j) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

According to Eq. (8), we may calculate the precision rate P, recall rate R and the F-measure F of the clustering results:

$$P = \mathrm{Avg}_i\left[\mathrm{Avg}_{j:U_c(j)=U_c(i)}\left[\mathrm{Correct}(i,j)\right]\right] \tag{9}$$

$$R = \mathrm{Avg}_i\left[\mathrm{Avg}_{j:U_t(j)=U_t(i)}\left[\mathrm{Correct}(i,j)\right]\right] \tag{10}$$

$$F = \frac{2PR}{P+R} \tag{11}$$

4.2. Experiment settings

In the experiments, we design the RGCN-16 with four RGCN blocks and each RGCN block has four hidden layers. In Section 3, we construct the sub-graph for every node and the indirect number h needs to be decided. We test different values of h and find that $h = 2$ will lead to a good clustering result and $h \geq 3$ does not obtain better results. Hence there are only three hyper-parameters in the proposed approach: the direct nearest neighbors' amount $k1$, the single indirect nearest neighbors' amount $k2$, and the nearest neighbors' amount u for selecting edges.

In the training phase, we set a large $k1 = 200$ in order to expect more information to be back-propagated. And we select a small value $k2 = 10$ for the purpose of avoiding sub-graph being too large. To make sure that there is at least one edge for every two-hop node, we set $u = 10$. In the testing phase, we conduct different experiments on IJB-B-512 to investigate how $k1$ and $k2$ influence the performance. Fig. 4 gives the experimental results. Fig. 4(a) shows that the F-measure increases when $k1$ become larger, because larger $k1$ brings more candidate links to be predicted. We observe in Fig. 4(b) that larger $k2$ also brings higher F-measure. However, the performance reaches saturation when $k1$ and $k2$ increase to a certain value. So $k1$ and $k2$ should not be too large, if we take both efficiency and time into consideration. Through experiments, we find that the best results can be obtained when $k1 = 80$. We also test how different $k2$ values influence the experimental results, and we find that $k2 = 7$ can achieve good results. For the parameter u, we set $u = k2$ to ensure every 2-hop node has at least one edge. Finally, we set $k1 = 80$, $k2 = 7$, $u = 7$ in the following experiment.

Table 2
Comparison of F-measure and NMI of different algorithms.

Method	Dataset					
	IJB-B-512		IJB-B-1024		IJB-B-1845	
	F	NMI	F	NMI	F	NMI
K-means [3]	0.612	0.858	0.603	0.865	0.600	0.868
Spectral [4]	0.517	0.784	0.508	0.792	0.516	0.785
AHC [11]	0.795	0.917	0.797	0.925	0.793	0.923
DBSCAN [5]	0.753	0.841	0.725	0.833	0.695	0.814
AP [32]	0.494	0.854	0.484	0.864	0.477	0.869
PAHC* [12]	–	–	0.639	0.890	0.610	0.890
ARO [1]	0.763	0.898	0.758	0.908	0.755	0.913
ConPaC* [33]	0.656	–	0.641	–	0.634	–
DDC [13]	0.802	0.921	0.805	0.926	0.800	0.929
SDCN [34]	0.474	0.836	0.452	0.830	–	–
LBF* [7]	0.852	0.937	0.855	0.944	0.857	0.958
RGCN-16 (ours)	**0.878**	**0.945**	**0.885**	**0.956**	**0.911**	**0.974**

4.3. Evaluation

The proposed deep face clustering approach RGCN-16 is compared with 11 clustering algorithms including traditional and state-of-the-art clustering methods: K-means [3], Spectral Clustering [4], Agglomerative Hierarchical Clustering (AHC) [11], DBSCAN [5], Affinity Propagation (AP) [32], Proximity-Aware Hierarchical Clustering (PAHC) [12], Approximate Rank-Order Clustering (ARO) [1], Conditional Pairwise Clustering (ConPaC) [33], Deep Density Clustering (DDC) [13], Structural Deep Clustering Network (SDCN) [34] and Linkage-Based Face clustering algorithm (LBF) [7].

(1) Clustering comparison. We compare the clustering results of the above algorithms on different datasets using F-measure and NMI. The comparison results are shown in Table 2 and the best results are marked in bold. The clustering results labeled asterisk (*) are come from the original paper. Table 2 is separated into three parts. The first part lists the clustering results of five classic clustering algorithms, among which AHC performs well on benchmark datasets. AHC is a bottom-up clustering method that can detect the number of clusters automatically according to the distance of data points and clusters. But AHC is easy to be influenced by the cluster distance threshold. The second part of Table 2 presents the clustering results of six state-of-the-art clustering algorithms: PAHC, ARO, ConPaC, DDC, SDCN and LBF. Benefit from the advantages of GCNs, the clustering results of LBF are much better than other clustering algorithms. LBF has no clustering assumptions about the distribution of datasets and it can predict the linkage relationship of graph nodes using the local context information.

The third part of Table 2 is our proposed method RGCN-16. The experiments show that RGCN-16 achieves the best clustering results on each dataset. RGCN-16 constructs a deep GCN clustering framework to improve the clustering quality. It solves the training problem of the original deep GCNs by introducing residual connections. Since the network structure is deeper, RGCN-16 can extract more graph features for clustering and produce good clustering results.

(2) Efficiency analysis. The runtime of the training and testing process of RGCN-16 increases linearly with the amount of data. Similar to GCN clustering, the most time-consuming part of RGCN-16 is constructing sub-graphs whose complexity hinges on the number of nodes. The computation complexity of sub-graph construction is $O(n\log n)$ if we seek the nearest neighbors using Approximate Nearest Neighbor (ANN) search. Another way to find the nearest neighbors is brute force search with complexity $O(n^2)$. In order to intuitively know the time consumption of the two algorithms, we call *pyflann* library to implement sub-graph

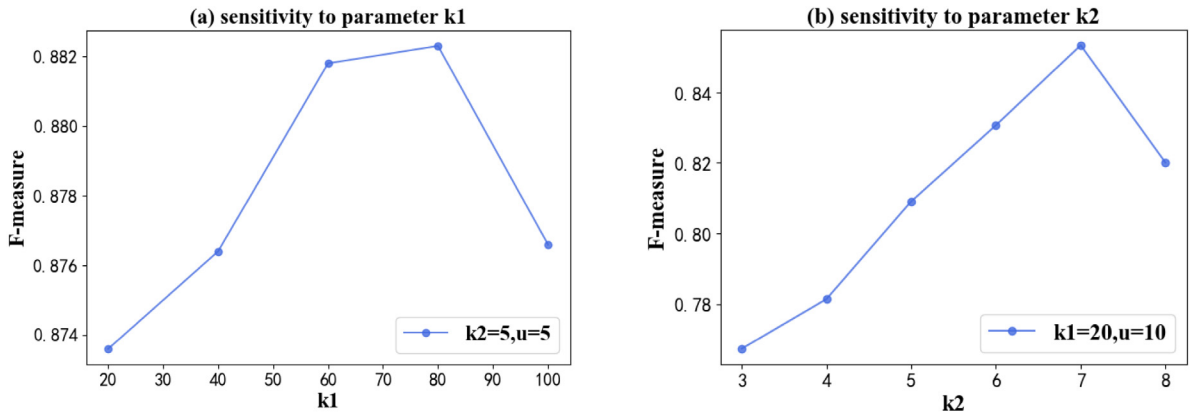

Fig. 4. Clustering results of the proposed framework on IJB-B-512. (a) The F-measure varies with $k1$, with constants $k2 = 5$ and $u = 5$. (b) The F-measure varies with $k2$, with constants $k1 = 20$ and $u = 10$.

Table 3
Comparison of two sub-graph construction methods used in RGCN-16.

Method	Dataset								
	IJB-B-512			IJB-B-1024			IJB-B-1845		
	F	NMI	Runtime	F	NMI	Runtime	F	NMI	Runtime
ANN	0.878	0.945	2.026 s	0.885	0.956	4.239 s	0.911	0.974	9.367 s
Brute force	0.885	0.948	71.875 s	0.896	0.961	284.789 s	0.919	0.977	931.183 s

construction on three datasets and record the runtime. *Pyflann* is the python bindings for FLANN — Fast Library for Approximate Nearest Neighbors. We test ANN search and brute force search on GPU and the runtime of the two methods is given in Table 3. The influence of the two sub-graph construction methods on the clustering performance of RGCN-16 is also different, and the clustering results can be found in Table 3. Compared with ANN, brute force search brings higher precision and recall rate, and thus leads to relatively higher NMI and F-measure score. But the performance improvement of brute force search is not obvious, and its running time is very long due to the high computational complexity. Taking scalability and efficiency into consideration, we prefer to use ANN search in RGCN-16. In general, the overall complexity of the proposed RGCN clustering method is $O(n\log n)$.

5. Conclusion and future work

GCNs have proven to be a useful tool in solving face clustering problems. In this paper, we try to improve the performance of GCN clustering by constructing a deep GCN architecture. However, training deep GCNs suffers from the vanishing gradient problem, which is a bottleneck in current GCN clustering research. To overcome this problem, we propose the RGCN clustering approach based on residual graph convolutional networks. RGCN stacks multiple layers into GCNs and has stronger learning ability. The proposed clustering approach constructs a sub-graph for each node with its nearest neighbors, and utilizes RGCN to learn the context information from these sub-graphs to do clustering. RGCN outputs the possibility that the central node is linked to its nearest neighbors, and we merge the most likely connected nodes to get a cluster. The effectiveness of the proposed RGCN clustering approach is tested on benchmark datasets. Comprehensive experiments indicate that our method is superior to other popular clustering algorithms on large-scale face datasets.

In the future, we will study how to extend the RGCN clustering framework deeper or wider, or use multi-view technique to further improve the clustering quality. It will also be interesting to apply the ideas of DenseNet and dilated convolution in CNNs to GCNs. We expect RGCN to become a powerful tool not only in clustering tasks but also in other areas of artificial intelligence.

CRediT authorship contribution statement

Chao Qi: Conceptualization, Methodology, Writing - original draft. **Jianming Zhang:** Writing - review & editing. **Hongjie Jia:** Supervision, Writing - review & editing. **Qirong Mao:** Writing - review & editing. **Liangjun Wang:** Software, Data curation, Validation. **Heping Song:** Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Key projects of the National Natural Science Foundation of China (No. U1836220), the National Natural Science Foundation of China (Nos. 61906077, 61672267, 61601202), the Natural Science Foundation of Jiangsu Province (Nos. BK20190838, BK20170558), the Project funded by China Postdoctoral Science Foundation (Nos. 2020M671376, 2020T130257), and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Nos. 18KJB520009, 16KJB520008).

References

[1] C. Otto, D. Wang, A.K. Jain, Clustering millions of faces by identity, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2) (2017) 289–303.
[2] Z. Liu, Y.Q. Song, C.H. Xie, Z. Tang, A new clustering method of gene expression data based on multivariate Gaussian mixture models, Signal Image Video Process. 10 (2) (2016) 359–368.
[3] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (2) (1982) 129–137.
[4] L. Wang, S. Ding, H. Jia, An improvement of spectral clustering via message passing and density sensitive similarity, IEEE Access 7 (2019) 101054–101062.
[5] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining (Vol. 96, (34) 226-231), 1996.

[6] M. Zhang, Y. Chen, Link prediction based on graph neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 5165–5175.
[7] Z. Wang, L. Zheng, Y. Li, S. Wang, Linkage based face clustering via graph convolution network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1117–1125.
[8] Q. Li, Z. Han, X. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3538–3545.
[9] G. Li, M. Muller, A. Thabet, B. Ghanem, DeepGCNs: Can GCNs go as deep as CNNs? in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9267–9276.
[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[11] C. Zhu, F. Wen, J. Sun, A rank-order distance based clustering algorithm for face tagging, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 481–488.
[12] W.A. Lin, J.C. Chen, R. Chellappa, A proximity-aware hierarchical clustering of faces, in: Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition, 2017, pp. 294–301.
[13] W.A. Lin, J.C. Chen, C.D. Castillo, R. Chellappa, Deep density clustering of unconstrained faces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8128–8137.
[14] H. Zhu, C. Chen, L.Z. Liao, M.K. Ng, Multiple graphs clustering by gradient flow method, J. Franklin Inst. B 355 (4) (2018) 1819–1845.
[15] M. Tapaswi, M.T. Law, S. Fidler, Video face clustering with unknown number of clusters, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5027–5036.
[16] L. Yang, X. Zhan, D. Chen, J. Yan, C.C. Loy, D. Lin, Learning to cluster faces on an affinity graph, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2298–2306.
[17] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[18] X. Zhou, J. Sun, Y. Tian, X. Wu, C. Dai, B. Li, Spectral classification of lettuce cadmium stress based on information fusion and VISSA-GOA-SVM algorithm, J. Food Process Eng. 42 (5) (2019) e13085.
[19] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, 2017, pp. 1–14.
[20] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[21] T. Zhan, R. Yu, Y. Zheng, Y. Zhan, L. Xiao, Z. Wei, Multimodal spatial-based segmentation framework for white matter lesions in multi-sequence magnetic resonance images, Biomed. Signal Process. Control 31 (2017) 52–62.
[22] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[23] Z. Liu, Y.Q. Song, V.S. Sheng, L. Wang, R. Jiang, X. Zhang, D. Yuan, Liver CT sequence segmentation based with improved U-Net and graph cut, Expert Syst. Appl. 126 (2019) 54–63.
[24] X. Zhan, Z. Liu, J. Yan, D. Lin, C. Change Loy, Consensus-driven propagation in massive unlabeled data for face recognition, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 568–583.
[25] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.
[26] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 87–102.
[27] Q. Cao, L. Shen, W. Xie, O.M. Parkhi, A. Zisserman, Vggface2: A dataset for recognising faces across pose and age, in: Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition, 2018, pp. 67–74.
[28] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, 2014, arXiv preprint arXiv:1411.7923.
[29] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, … J. Cheney, Iarpa janus benchmark-b face dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 90–98.
[30] H. Jia, S. Ding, M. Du, Y. Xue, Approximate normalized cuts without Eigen-decomposition, Inform. Sci. 13 (2016) 5–150.
[31] X. Wu, B. Wu, J. Sun, S. Qiu, X. Li, A hybrid fuzzy K-harmonic means clustering algorithm, Appl. Math. Model. 39 (12) (2015) 3398–3409.
[32] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[33] Y. Shi, C. Otto, A.K. Jain, Face clustering: representation and pairwise constraints, IEEE Trans. Inf. Forensics Secur. 13 (7) (2018) 1626–1640.
[34] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, P. Cui, Structural deep clustering network, in: Proceedings of the International World Wide Web Conference, 2020, pp. 1400–1410.