Face Clustering via Graph Convolutional Networks with Confidence Edges

Yang Wu[1,3,4]* Zhiwei Ge[2] Yuhao Luo[2]† Lin Liu[2] Sulong Xu[2]
[1] Institute of Automation, Chinese Academy of Sciences
[2] JD.COM [3] SKLCS, Institute of Software, Chinese Academy of Sciences
[4] University of Chinese Academy of Sciences, Beijing, China
wuyang2023@ia.ac.cn {gezhiwei,luoyuhao6,liulin1,xusulong}@jd.com

Abstract

Face clustering is a method for unlabeled image annotation and has attracted increasing attention. Existing methods have made significant breakthroughs by introducing Graph Convolutional Networks (GCNs) on the affinity graph. However, such graphs will contain many vertex pairs with inconsistent similarities and labels, thus degrading the model's performance. There are already relevant efforts for this problem, but the information about features needs to be mined further. In this paper, we define a new concept called confidence edge and guide the construction of graphs. Furthermore, a novel confidence-GCN is proposed to cluster face images by deriving more confidence edges. Firstly, Local Information Fusion is advanced to obtain a more accurate similarity metric by considering the neighbors of vertices. Then Unsupervised Neighbor Determination is used to discard low-quality edges based on similarity differences. Moreover, we elaborate that the remaining edges retain the most beneficial information to demonstrate the validity. At last, the confidence-GCN takes the graph as the input and fully uses the confidence edges to complete the clustering. Experiments show that our method outperforms existing methods on the face and person datasets to achieve state-of-the-art. At the same time, comparable results are obtained on the fashion dataset.

1. Introduction

In recent years, and thanks to existing works[24, 25], face recognition has rapidly developed and is now widely used in face verification, security systems, and other daily applications. This development relies on high-quality annotated data, but manual annotation is extremely expensive and time-consuming. Therefore, face clustering emerges and becomes a primary technology to address the problem.

Face clustering has gained considerable developments recently. Existing face clustering methods can be divided into two categories: unsupervised and supervised [14, 4, 32, 7, 28, 27]. K-means [14], and DBSCAN [4] are representative unsupervised methods but perform poorly due to their limited capacities. Recently, a new unsupervised method FaceMap[35] has achieved positive results by adjusting the relationships between images to fit traditional clustering methods. Although this is very effective, original features are not clustering-oriented, so the performance will undoubtedly be subject to certain limitations.

Supervised learning methods [7, 33, 32, 21, 27] are mainly based on graph convolution networks (GCNs) [30]. Such methods first build face graphs by deeming images as vertices and then linking them based on their deep features, which are extracted from a trained Convolutional Neural Network (CNN) [11]. In existing research [33, 32], such graphs are often called affinity graphs. The most common affinity graphs are built based on kNN [3] (k-nearest neighbors) relations, where each vertex is connected to its top k neighbors. With affinity graphs as input, a GCN can be utilized to capture the structural information and embed it into the features. Due to its powerful feature propagation capability, the clustering performance is significantly improved.

However, several unfavorable factors limit GCN-based models. First, since the training and inference of CNN models treat each image as an individual without considering their associations, images of different identities may have high similarity. The opposite may also occur in the same identities. Hence, affinity graphs would inevitably contain many undesirable edges whose endpoints have incompatible labels and similarities. GCN will propagate harmful information through such edges and obtain polluted features, hindering performance. Second, real datasets often contain inaccurate labels. Even if the graph is clean enough, untrusted labels will lead the GCN to learn incorrect information, which is not conducive to clustering.

To amend these drawbacks, we combine labels with similarities to adjust face graphs and propose a novel GCN

*This work was done when Yang Wu was an intern at JD.COM
†Corresponding author

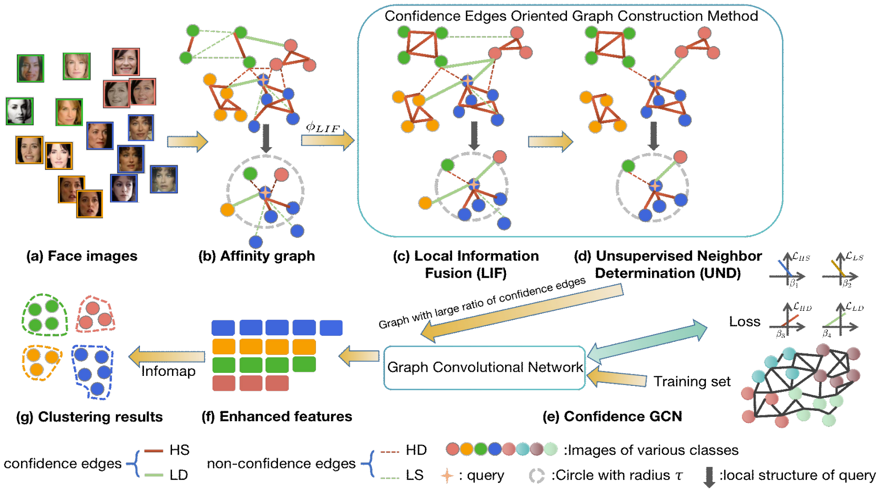

Figure 1. The framework of our method. Given face images (a), We first build affinity graphs (where each vertex is connected to k nearest neighbors (b)) based on the cosine similarity of the features. There exist numerous non-confidence edges in the graph. To get more confidence edges, we first use LIF (c) to derive a more reliable similarity metric ϕ_{LIF} between vertices. Then, based on the new metric, UND (d) is used to find suitable neighbors for each vertex and build a graph. As UND discards neighbors based on similarity, hard examples will remain that need further treatment. A novel GCN model confidence-GCN (e) takes the graph as input to get enhanced features (f) and promote confidence. In the end, the traditional clustering method Infomap[19] achieves the clustering results (g).

model to alleviate the influence of label errors, thereby boosting the clustering performance. The underlying principle is the more consistent the labels and similarities of the vertices are (i.e., vertices of the same (different) labels have a high similarity (difference) score), the better the performance will be. To formalize it, we propose the concept of **confidence edges** to denote the highly consistent edges. Specifically, given an affinity graph $G = (V, E)$, its adjacent matrix $A = (a_{ij})$, and a threshold τ, let $H = \{(v_i, v_j) \in E | v_i, v_j \in V, a_{ij} \geq \tau\}$ be the set of high similarity edges, and $L = E \setminus H$ is the set of low similarity ones. Meanwhile, we can partition E into S and D based on labels, where S is the set of edges whose endpoints have the same labels and D is the others. Cross combines similarities and labels to get the following four types: HS, LS, HD, and LD, where HS and LD form confidence edges. As illustrated in Figures 1, many non-confidence edges exist in affinity graphs (b) whose similarity of endpoints can not reflect their categories. Thus, increasing the number of confidence edges will advance clustering performance.

To obtain more confidence-edges, we first re-evaluate the tightness between vertex pairs. The **local information fusion** (LIF) is proposed to mine the local information of the vertices and provide a more precise metric. For every vertex pair, LIF exploits the quantity and similarity differences between common neighbors to improve the cosine similarity. The new metric will contain sufficient contextual information to compensate for the weakness of the features. Consequently, the closeness of vertices in the same class increase with similar neighborhood structures, while others reduce.

Next, we propose **unsupervised neighbor determination** (UND) to adjust the face graphs and increase the ratio of confidence edges. Existing kNN method cannot meet the need of each vertex. Many valuable edges will be discarded when k is small . However, if k is large , there will undoubtedly be numerous non-confidence edges. The UND exploits the neighborhood statistics of every vertex in the affinity graph to filter out the most similar parts and get rid of the rest. With UND, a graph containing many confidence edges is derived. Better yet, the impact of incorrect labels eliminates since UND avoids using labels.

Then, GCN takes the graphs as input to increase the similarity (difference) between positive (negative) pairs and outputs features beneficial for clustering. To further increase the quantity of the confidence edges, we propose **confidence-GCN**, which gives a higher level of attention to the non-confidence edges and accelerates to narrow the gap between the labels and similarities. During training, the intra-class similarities and inter-class differences increase, which yields more confidence edges and speeds up convergence, forming a virtuous circle. Under the dual guidance of labels and similarities, the model can effectively deal with label errors and output better features to finish clustering.

The contributions of our paper are as follows.

- We propose **confidence edges** to guide the building of face graphs. And a novel con-GCN is proposed to fully utilize confidence edges and raise their share, thus significantly improving the performance.

- LIF and UND are proposed to increase the ratio of confidence edges to yield an improved face graph. Furthermore, UND is detailed using information theory to prove its effectiveness.

- Our approach achieves state-of-the-art performances on faces and person re-identification. Besides, a comparable result is obtained on closets.

2. Related Work

2.1. Face Clustering

In recent years, face clustering has attracted considerable attention as an essential machine-learning task. Traditional clustering methods like K-means[14] and DBSCAN[4] provide highly explanatory but perform poorly. Since K-means require assumptions about the dataset, DBSCAN cannot cope with high-dimensional situations. Until recently, GCN-based methods have been proposed and dramatically improved performance. L-GCN[28] extracts affinity graphs into several Instance Pivot Subgraphs and trains a GCN to predict links on subgraphs. GCN-D+S [33] uses GCN-D to generate subgraphs and GCN-S to cluster face images. GCN-V+E [32] also adopts two GCNs to predict vertices' confidence and link relationships on affinity graphs. STAR-FC[21] samples structure-preserved subgraphs and finishes cluster by a GCN. FaceMap[35] views face clustering as community detection and adjusts the affinity graph to fit the traditional method. Pair-Cls[12] employs density to select image pairs and train a pairwise classification model to cluster face. MHC[2] develops a size-invariant density NDDe and sparsity-aware distance TPDi to improve the performance of a traditional method DPC[18]. Ada-NETS[27] proposes structure space and adaptive neighbor discovery to remove noise edges whose two endpoints belong to different classes, then gets more accurate clustering results. However, Ada-NETS proposed noise edges cannot wholly judge the quality of edges. Such as, edges with different labels and low similarity belong to noise edges but are also beneficial to clustering.

2.2. Graph Convolutional Network

Graphs are widely used in social networks, recommender systems[34, 15], and other domains. The classical models CNN and RNN cannot handle such non-Euclidean data. Graph convolutional network(GCN)[30] emerges and performs very well. However, GCN in a transductive setting has poor flexibility and is difficult to expand. GraphSAGE[9] extends transductive learning to inductive learning by sampling a fixed number of neighbors. GAT[26] introduces attention mechanism, taking the correlation between vertices into account. GAAN[36] makes incremental improvements over GAT by computing an additional attention score for each attention head. SGC[31] uses the K-th power of the graph revolution matrix to capture k-hop neighbor information to simplify the model. The above models yield outstanding results on plenty of essential tasks. However, due to the lack of natural graph structure, the face graphs rely on the relationship between features of vertices. Many edges in the graphs cannot correctly reflect the relationship between vertices. Moreover, the mentioned GCN models cannot adequately deal with these edges and propagate the useless information between vertices, thus affecting the effect.

3. Method

In this section, we describe the proposed face clustering method in detail. The method is outlined in Figure 1. Given large-scale face images, the deep features $\mathcal{V} = \{v_1, v_2, \cdots, v_N | v_i \in \mathbb{R}^D\}$ are extracted by a pretrained CNN model. The LIF module provides a more accurate similarity metric, which is then used by the UND module to find suitable neighbors for each vertex. A graph with a large proportion of confidence edges will be built. The confidence GCN takes this graph as the input increases the intra-class similarity and inter-class differences. The output features can be utilized in various traditional clustering methods to achieve clustering results.

3.1. Confidence Edges Oriented Graph Construction

Non-confidence edges in face graphs lead the GCN to aggregate erroneous information when propagating messages. The existence of such edges can be attributed to the following reasons. First, limited by the representation ability of CNN, images of different classes may have high similarity. Hence, graphs built based on the similarities of deep features will bring in undesirable LS and HD edges (as outlined in Section 1). Second, neither the kNN methods [21, 32, 33] nor the threshold methods [7] can accurately satisfy each vertex, resulting in numerous non-confidence edges. Third, real datasets may include label errors, which are difficult to detect. To address these challenges, we first derive a more accurate similarity metric by fusing the local information and then determine suitable neighbors for each image in an unsupervised way.

3.1.1 Local Information Fusion(LIF)

Feature similarities cannot accurately reflect the relationship between images, but neighborhood information is capable of helping alleviate this problem. The idea stems from SS[27], who uses the Jaccard index to improve Cosine similarity. And its validity proves the benefit of the idea.

However, hard cases, as shown in Figure 2, can not be effectively solved. Compared with v_1 and v_2, v_1 and v_3 of different classes have higher cosine similarity and the same Jaccard index. Further observation reveals that v_1 and v_2 have closer common neighbors, it helps v_1 find true peers.

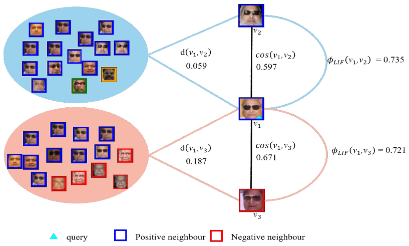

Figure 2. Cosine similarity and the Jaccard index cannot correctly reflect the label relationships between vertices. LIF can use similarity differences to give correct results. The right side shows the similarities with $\lambda = 0.6$ and $\mu = 0.8$.

Based on this idea, we propose LIF and derive a more discriminative similarity metric. Given any images $v_i \in \mathcal{V}$, $\mathcal{N}_i$ is the k nearest neighbor sequences by descending order of cosine similarity. And $\mathcal{N}_{ij} = \mathcal{N}_i \bigcap \mathcal{N}_j$ is common neighbors of v_i, v_j. The similarity of v_i, v_j is defined as:

$$\phi(v_i, v_j) = \mu cos(v_i, v_{i_j}) + (1 - \mu) s(v_i, v_j) \tag{1}$$

$$s(v_i, v_j) = \gamma Jac(v_i, v_j) + (1 - \gamma)(1 - d(v_i, v_j)) \tag{2}$$

$$d(v_i, v_j) = \frac{1}{|\mathcal{N}_{ij}|} \sum_{l \in \mathcal{N}_{ij}} |cos(v_i, v_l) - cos(v_j, v_l)| \tag{3}$$

where $cos(v_i, v_j)$ and $Jac(v_i, v_j) = \frac{|\mathcal{N}_{ij}|}{|\mathcal{N}_i \bigcup \mathcal{N}_j|}$ denote the cosine and Jaccard similarities, respectively. $d(v_i, v_j)$ is the similarity differences. $\mu, \gamma \in [0, 1]$ are the weights.

Equation (3) contains rich structural information. In the induced subgraph $G[\mathcal{N}_{ij} \cup \{v_i, v_j\}]$, the more minor $d(v_i, v_j)$ is, the more similar the structural roles (i.e., structural equivalence [6]) of v_i and v_j are. As referred to in node2vec [6], v_i and v_j should be more similar. With the help of LIF, the similarity metric yields more confidence.

3.1.2 Unsupervised Neighbor Determination(UND)

However, due to the different requirements of each vertex, a graph built based on a given k or threshold will still be unsatisfactory. AND [27] considers the operation of linking edges as a sequence task and applies Long Short-Term Memory (LSTM) [20, 5] to abandon neighbors that lower the F_β-score. However, there are two limitations. First, the sequence property is insufficient as the adjacent points are independent. Second, maximizing the F_β-score is not always reasonable, especially in real datasets where error labels may exist.

To compensate for these deficiencies, we propose UND. The key idea is that, for any image v_i, the region with the fastest decrease in similarity is an ideal neighborhood boundary. With LIF, all vertices after the boundary are less similar with v_i in feature and structure. Therefore, discarding them will remove numerous non-confidence edges.

We will discuss UND from the perspective of information theory. Given v_i, the k nearest neighbors in descending order of similarity $\mathcal{N}(v_i) = \{v_{i_1}, \cdots, v_{i_k}\}$. Moreover, let $\mathcal{S}_i$ be the corresponding similarity sequence. The information entropy increases when linking the edges one by one from v_{i_1}. And v_{i_j} would provide $-p_{ii_j} log_2 p_{ii_j}$ to entropy, where $p_{ii_j} = \frac{\phi_{LIF}(v_i, v_{i_j})}{\sum \phi_{LIF}(v_i, v_{i_l})}$ is the transition probability from v_i to v_{i_j}. The larger the similarity, the greater the information gain. The boundary is drawn when the amount of information reduces sharply. So not only major information retained but also meaningless edges are excluded.

From another perspective, looking from the back of the sequence, the boundary is where the similarity decreases fastest relative to the tail. For this relativity, we adopt the Z-score of the first-order difference sequence to represent the falling speed. Besides, there are possible abnormal points whose similarities drop suddenly and cannot reflect the surrounding situation. To avoid the influence of abnormal points, motivated by the Harris corner detection [10], we utilize the mean value of an interval to approximate the Z-score of the midpoint. For v_i, the first-order difference sequence $\Delta\mathcal{S}_i(j) = \mathcal{S}_i(j) - \mathcal{S}_i(j + 1)$. The Z-score of $v_{i_j} \in \mathcal{N}(v_i)$ relative to the tail is

$$z_i(i_j) = \frac{\sum\limits_{l=i_j-\lfloor \mathcal{I}/2 \rfloor}^{i_j+\lfloor \mathcal{I}/2 \rfloor} \Delta\mathcal{S}_i(l) - \mu_{i_j}}{\sigma_{i_j}} \tag{4}$$

where μ_{i_j} and σ_{i_j} are the mean and standard deviation of $\Delta\mathcal{S}_i$ in $[i_j + \lfloor \mathcal{I}/2 \rfloor, k]$ respectively. $\mathcal{I}$ is the length of the interval.

The index corresponding to the maximum Z-score is taken as the boundary B_i (i.e., $B_i = \mathrm{argmax}_{i_j}\, z_i(i_j)$), and only linking the neighbors before the line. Graph G is constructed by performing the above operations on all vertices.

3.2. The Confidence-GCN Model(con-GCN)

Although LIF and UND increase the proportion of confidence edges, non-confidence edges still exist and need further treatment. Since the confidence edges are already sufficiently suited for clustering. Existing GCN does not fully utilize confidence ones by paying equal attention on all edges. Hence, we propose a novel confidence-GCN (con-GCN) model to take full advantage of the confidence edges.

Let A and $F = [v_1, v_2, \cdots, v_N]^T \in \mathcal{R}^{N \times d}$ be the adjacency and feature matrices of G, respectively. The proposed con-GCN consists of two layers, any of which is defined as

$$F_{l+1} = \sigma(\widetilde{A} F_l W_l) \tag{5}$$

where $\widetilde{A} = \widetilde{D}^{-1}(A + I)$, $\widetilde{D}$ is the diagonal degree matrix with $\widetilde{D}_{ii} = \sum_{j=1}^{N}(A+I)_{ij}$. F_l represents the embedding at the l-th layer, and F_0 is the input feature matrix F. W_l is a learnable matrix that converts the embeddings to a new space. $\sigma(\cdot)$ is a nonlinear activation function. The model finishes with a fully connected layer with PReLU activation. The output is an enhanced feature $v_i^{'}$ of each vertex v_i.

We pay more attention to the non-confidence edges to eliminate their adverse effects. As the iteration proceeds, the features are continuously enhanced. And the similarities become more consistent with labels, leading to a gradual increase in the ratio of confidence edges.

Besides, as LIF accurately evaluates the association between vertices and UND retains highly similar edges. Many LD edges' similarities reduce to zero, simplifying the graph.

In con-GCN, a new loss function guided by confidence edges is proposed, based on variant Hinge losses [27]:

$$\begin{aligned}
&\mathcal{L} == (\mathcal{L}_{HS} + \mathcal{L}_{LD}) + \lambda(\mathcal{L}_{HD} + \mathcal{L}_{LS}) \\
&\mathcal{L}_h = \frac{1}{|\mathcal{E}_h|} \sum_{(v_i,v_j)\in\mathcal{E}_h} 1 - s_{ij}^{'}\, , h \in \{HS, HD\} \\
&\mathcal{L}_l = \max_{(v_i,v_j)\in\mathcal{E}_l} 1 + s_{ij}\, , \qquad\qquad l \in \{LD, LS\}
\end{aligned} \tag{6}$$

Where $\mathcal{E}_i$, and $\mathcal{L}_i(i \in \{HS, LS, HD, LD\})$ are the set of edges and losses of the corresponding edge type, respectively. $s_{ij}^{'}$ is the cosine similarity between the enhanced features $v_i^{'}$ and $v_j^{'}$. λ balances different losses.

Given a threshold th, the variant Hinge losses [27] corresponding to four edge types are calculated separately. Furthermore, the weighted sum of the four losses is the final loss. The proposed con-GCN prioritizes fitting the non-confidence parts and suppresses the influence of label errors. Eventually, every vertex learns a better feature.

During training, we strive to ensure that vertices with the same labels have a higher cosine similarity and vice versa.

During inference, the trained GCN model obtains the output features $[v_1^{'}, v_2^{'}, \cdots, v_N^{'}]^T \in R^{N\times D^{'}}$ for all vertices. The enhanced features have significant inter-class differences and intra-class similarities. Traditional clustering methods would be more effective. In this study, the clustering results are achieved by Infomap [19], a network clustering algorithm. To accelerate, we set θ as the threshold and only link the vertices whose similarities are larger than θ.

3.3. Complexity Analysis

On affinity graph $G = (n, m)$, where n, m is the number of vertices and edges, respectively. The affinity graph is built based on the kNN method, so it satisfies $m \leq nk$.

The time complexity of graph construction consists of two parts. LIF traverses the edges. With each iteration, LIF first sorts the neighbor sequences of endpoints in $O(klogk)$, then calculates the intersection, union, and similarity difference in $O(k)$. In a similar vein, for every vertex and corresponding neighbors, UND computes the mean and variance of a subsequence(whose length is at most $k - \frac{\mathcal{I}}{2}$) amounted to $k - \mathcal{I}$ times. Therefore, the complexity of LIF is $O(mklogk) = O(nk^2logk)$, and of UND is $O(nk^2)$. For con-GCN, the sparsity of the graph ensures that the computation time for each layer is $O(m) = O(nk)$[30].

Since the k is a fixed parameter and usually with $k \ll n$, the complexities of the above operations can all be deemed as O(n), reflecting the time advantage of our method.

4. Experiments

4.1. Datasets, Metrics and Implementation Details

We evaluate our method on three datasets. **MS-Celeb-1M** [8] is a large-scale face dataset with 5.8M images of 86K identities. Following the experimental protocol in [21, 27, 32], the dataset is divided into ten parts, each with 8.6K identities and related images. The ReID dataset **MSMT17** [29] contains 4101 classes and 126441 images. 32621 images of 1041 individuals are used for training, and the remaining images are for testing. **DeepFashion**[13], a closets dataset, has a training set of 25752 images of 3997 classes and a test set of 26960 images of 3984 categories.

Pairwise F-score(F_P) [22] and BCubed F-score(F_B) [1] are used to evaluate the clustering performance. Besides, we use confidence ratio (CR), the ratio of the number of confidence to non-confidence edges in graphs, to compare different graph construction methods. Also, we introduce average similarity ratio (ASR) and modularity [16, 17, 37] to compare different similarities. ASR is the ratio of positive pairs' average similarity to negative pairs, and modularity is an important measure in community detection. It evaluates the strength of clustering results by computing the difference of compactness within clusters and separation between clusters for all clusters.

For MS-Celeb-1M, we set $k = 80$ for building affinity graph. The momentum SGD optimizer starts with a learning rate of 0.01 and is dynamically adjusted with a weight decay of 1e-5. Except k is 20 for DeepFashion, and k is 60 for MSMT17. And the learning rate is 0.001 for them. The rest settings are the same as that of MS-Celeb-1M.

4.2. Method Comparison

We compare our method with existing methods, including the unsupervised clustering methods K-Means[14], HAC[23], DBSCAN[4], and Facemap[35], and the supervised methods L-GCN[28], GCN-(V+E)[32], STAR-FC[21], MHC[2], and Ada-NETS[27]. The comparison results in Tables 1 and 2 demonstrate that our method achieves superior performance. Experiments on MS-Celeb-1M also highlight that the more unlabeled images there are, the more significant the performance improvement our proposed method can provide. When the data size is up to

#unlabeled	584k		1.74M		2.89M		4.05M		5.21M	
Method	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B
K-means[14]	79.21	81.23	73.04	75.2	69.83	72.34	67.9	70.57	66.47	69.42
HAC[23]	70.63	70.46	54.40	69.53	11.08	68.62	1.40	67.69	0.37	66.96
DBSCAN[4]	67.93	67.17	63.41	66.53	52.50	66.26	45.24	44.87	44.94	44.74
L-GCN[28]	78.68	84.37	75.83	81.61	74.29	80.11	73.70	79.33	72.99	78.60
GCN-V+E[32]	87.93	86.09	84.04	82.84	82.10	81.24	80.45	80.09	79.30	79.25
STAR-FC[21]	91.97	90.21	88.28	86.26	86.17	84.13	84.70	82.63	83.46	81.47
Ada-NETS[27]	92.79	91.40	89.33	87.98	87.50	86.03	85.40	84.48	83.99	83.28
FaceMap[35]	94.24	92.55	91.31	89.67	89.32	88.20	87.74	87.11	86.37	86.29
MHC[2]	93.22	92.18	90.51	89.43	89.09	88.00	87.93	86.92	86.94	86.06
Ours	**94.85**	**93.24**	**92.48**	**90.81**	**91.29**	**89.55**	**90.24**	**88.60**	**89.35**	**87.71**
Improvement	**+0.65%**	**+0.75%**	**+1.28%**	**+1.27%**	**+2.21%**	**+1.53%**	**+2.63%**	**+1.71%**	**+2.77%**	**+1.65%**

Table 1. Performance comparison with different numbers of unlabeled images on MS-Celeb-1M. Our model achieves state-of-the-art on all scales. Furthermore, the larger the scale, the more pronounced the improvement.

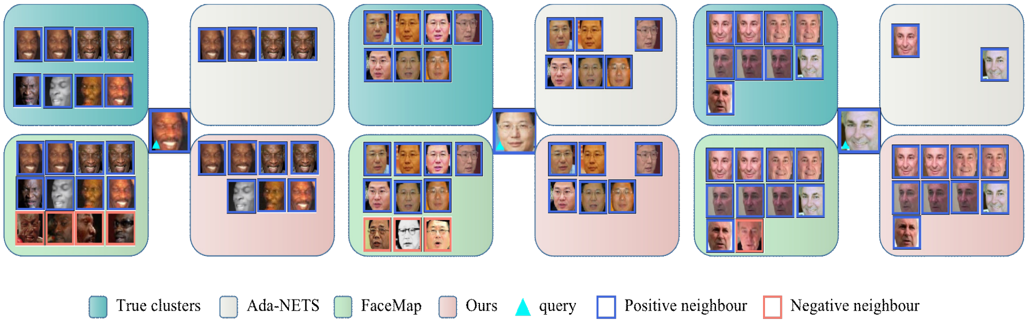

Figure 3. Random querys and corresponding results based on Ada-NETS[27], FaceMap[35], and our method. Ada-NETS has high precision but loses valuable peers, while Facemap guarantees a high recall, it introduces much noise. Our method can give attention to both, thus obtaining excellent results.

Datasets	MSMT17		DeepFashion	
Method	F_P	F_B	F_P	F_B
K-means[14]	53.82	62.41	32.86	53.77
HAC[23]	60.27	69.02	22.54	48.77
DBSCAN[4]	35.69	42.32	25.07	53.23
L-GCN[28]	49.19	62.06	28.85	58.91
GCN-V+E[32]	50.27	64.56	38.47	60.06
STAR-FC[21]	58.80	66.92	37.07	60.60
Ada-NETS[27]	64.05	72.88	39.30	61.05
FaceMap[35]	67.22	74.76	35.93	57.64
MHC[2]	-	-	40.91	**63.61**
Ours	**71.29**	**76.24**	**44.43**	63.28
Improvement	**+6.05%**	**+1.98%**	**+8.60%**	-0.52%

Table 2. Performance comparison on MSMT17 and DeepFashion. Our model performs far superior to the other models except for the F_B of the DeepFashion.

5.21M, the growth rate of F_P reaches 3.45%.

	ASR	$Modularity$
ϕ_{cos}	1.28	0.754
ϕ_{SS}	1.89	0.820
ϕ_{LIF}	**2.17**	**0.840**

Table 3. ASR and modularity scores of different similarity metrics on affinity graphs built based on k-NN (k is set to be 80). Where ϕ_{cos} denotes cosine similarity of features and ϕ_{SS} is the similarity metric after structure space[27].

More significant improvements are also witnessed in the MSMT17 and DeepFashion datasets. In MSMT17, our method reaches 71.29 on the F_P from 67.22, with an increase of 6.05%. In DeepFashion dataset, our approach outperforms the state-of-the-art by 3.52 for F_P and reaches a formidable 8.60%. The significant increase in the above datasets illustrates that our method can effectively improve clustering performance, especially for the Pairwise F-score.

Moreover, as shown in Figure 3, we randomly sample three queries to compare the actual results. The samples show that Ada-NETS loses many similar images to keep a high precision. While Facemap guarantees a high recall, it introduces much noise. Our approach takes both precision and recall into account to obtain excellent results.

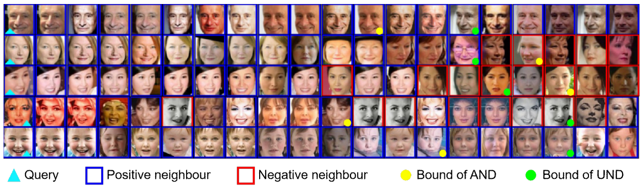

Figure 4. Random examples of the top 20 images ranked by similarity with query images are accompanied by the prediction values of UND and AND([27] 3.2). In most instances, UND can find better boundaries and then build higher-quality graphs.

4.3. Ablation Experiment

In this subsection, we select MS-Celeb-1M(584K) for the ablation study.

4.3.1 Study on Graph Construction Method

For a fair comparison, we first build kNN affinity graphs based on metrics ϕ_{cos}, ϕ_{SS}, ϕ_{LIF}. Where ϕ_{cos} is the cosine similarity, ϕ_{SS} and ϕ_{LIF} are similarity metric with structure space(SS)[27] and LIF, respectively. Then, the labels are used to ensure the graph vertices are perfectly clustered.

Building methods	CR	Time
affinity graph	1.63	67.64s
AND	9.24	262.15s
UND	8.05	62.70s
LIF+affinity graph	15.30	**56.13s**
LIF+AND	22.58	258.70s
LIF+UND	**27.52**	73.95s

Table 4. CR of different graph construction methods, where affinity graphs are built based on LIF and UND can both increase the ratio of confidence edges. Besides, the combination of both builds the best graph.

As shown in Table 3, both ϕ_{LIF} and ϕ_{SS} can enlarge the ASR by increasing the similarity of positive pairs and the difference of negative ones. Furthermore, the affinity graph with ϕ_{LIF} has a larger modularity score, meaning vertices within a class are closer, showing the advantages of LIF. So the neighborhood information improves the similarity measure, and LIF is better than SS, confirming our approach.

The actual results of UND and AND[27] are shown in Figure 4. Due to only utilizing the differences in the similarity between adjacent images, in some cases, UND may be confused by similar images from different people, resulting in poor results. However, in most cases, UND can find better boundaries, thereby reducing more non-confidence edges. Additionally, Table 4 shows that the time consumption of UND is much lower than AND (even excluding the training time).

To further explore the contributions of different modules, we compare the CR and time consumption of different combinations. The results are shown in Table 4. LIF, AND, and UND all contribute to eliminating non-confidence edges. Of the three, LIF contributes the most as a single module. AND obtains better results than UND when used alone. However, combining UND with LIF allows its capacity to be adequately exploited and obtain the most high-quality graph.

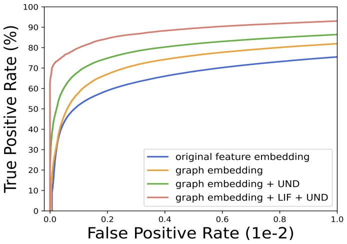

Figure 5. ROC curves for ten million randomly selected pairs of different feature embeddings. All graph embeddings are better than the original features. In addition, UND and LIF are capable of promoting con-GCN to generate more distinguishing features.

Meanwhile, the effects of different modules on the clustering results are shown in Table 5. With a more confident graph, the combination of LIF and UND yields better performance on all three datasets than on others.

4.3.2 Study on the confidence-GCN Model

To make a more comprehensive comparison of the proposed con-GCN, we conduct comparisons on different graphs. Table 5 shows that con-GCN achieves better clustering results on all construction methods and datasets, and this improvement is more remarkable on MSMT17 and DeepFashion.

Compared with commonly used GCN, con-GCN makes a crucial contribution to the performance of our method, as illustrated in Table 5. With con-GCN, the F_P and F_B on MS-Celeb-1M increase by more than 0.5. While on the MSMT17 and DeepFashion, the improvement is more prominent. Especially the F_P on MSMT17 increases by 4.64. Therefore, con-GCN can better increase the proportion of confidence edges. Additionally, since the same model has a lower F_P and F_B for MSMT17 and DeepFashion, it is reasonable to believe that these two datasets may contain more incorrect labels. The considerable improvement of con-GCN reflects its unique advantages when dealing with real datasets with possible label errors.

4.3.3 Study on the graph embedding

With the original features as input, con-GCN produces enhanced features that are more conducive to clustering. The ROC curves (Receiver Operating Characteristic curves) of ten million randomly selected pairs of different feature embeddings are shown in Figure 5. It is observed that graph

Method				MS-Celeb-1M		MSMT17		DeepFashion	
SS	LIF	UND	confidence GCN	F_P	F_B	F_P	F_B	F_P	F_B
×	×	×	×	89.06	86.93	56.05	63.29	34.17	52.73
×	×	×	✓	91.05	88.88	64.66	70.54	37.88	58.27
✓	×	✓	×	94.15	92.63	65.83	71.87	40.89	60.71
✓	×	✓	✓	94.71	93.09	70.40	75.79	43.38	62.69
×	✓	✓	×	94.36	92.79	66.65	73.22	41.06	61.24
×	✓	✓	✓	**94.85**	**93.24**	**71.29**	**76.24**	**44.43**	**63.28**

Table 5. The F_P and F_B value of the method of combining different modules on the MS-Celeb-1M. Compared with SS, LIF has mined more information. At the same time, the quality of the graph is greatly improved after being deleted by UND, making the effect better. Based on the importance of non-confident edges, con-GCN obtains better features than GCN, further improving the clustering effect.

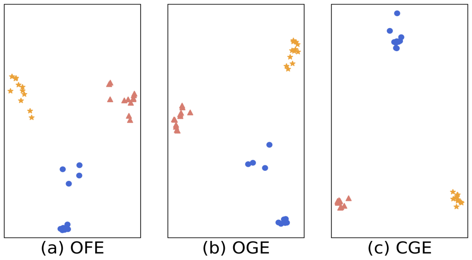

Figure 6. Distribution visualization for three different embeddings after dimensionality reduction by principal component analysis(PCA). Among them, the original feature embedding (OFE) (a) can distinguish three classes, but it is not good enough. The distance between different classes in embedding from the original GCN (OGE) (b) is larger than that of the original features. While con-GCN (CGE) (c) makes not only inter-class distance more significant but also the within-class data more intensive, reflecting the excellent performance of con-GCN embedding.

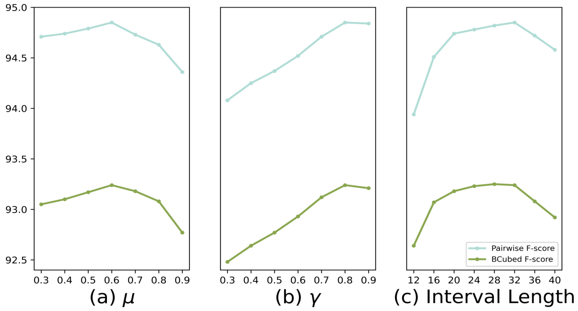

Figure 7. Inclueence of μ (Equation (1)), γ (Equation (2)),interval length on MS-Celeb-1M. With the change of μ(a), the evaluation results of clustering fluctuate slightly. In the same way, the change of the F-score of γ (b) is also gentle, although it is slightly steep compared with μ. Similarly, the curve of Interval length (c) is smooth in most ranges. Only when the value is too small or too large, it changes dramatically. The results show that our method is not sensitive to parameters reflecting the robustness of the model.

embeddings are better than the original features by a large margin. Besides, UND will increase the ratio of confidence edges and enhance the graph embedding. Furthermore, with the help of LIF, such embedding can be improved further, thus achieving the best cluster performance.

For the purpose of better reflecting the impact of different features on intra-class similarity and inter-class differences, we exploit principal component analysis to reduce the dimensions of different embeddings. As illustrated in Figure 6, compared with the original features (OFE), the embedding output of GCN (OGE) can increase the distance between classes, making the clustering effect more obvious, but the intra-class compactness is still insufficient. Con-GCN further features by putting more attention on the non-confidence edges. So the differences between classes and the closeness within class become more prominent.

4.4. Sensitivity Analysis

As illustrated in Figure 7, the model is robust to its parameters. In Figure 7 (a), although μ ranges from 0.3 to 0.9, the F-score changes small, with the maximum difference of F_P being 0.5. A similar pattern is reflected with γ in Figure 7 (b). But compared to μ, the curve changes slightly steeper. The interval length has the ability to avoid possible errors in Z-value calculation. At the same time, the change of interval length has little effect on the performance of the model. As shown in Figure 7 (c), performance robustness will be obtained within the appropriate value range. However, errors are introduced when the interval reaches 12, resulting in poor performance. Besides, if the interval is large, the mean of the interval can not accurately approximate the intermediate point, which also affects the clustering.

5. Conclusion

In this paper, we propose a new concept called confidence edges to select graphs for clustering. And a novel GCN-based face clustering method is proposed under the guidance of confidence edges. The confidence edges-oriented graph construction method contains two closely related modules: local information fusion and unsupervised neighbor determination. The former mines neighborhood information and reconstructs the similarity metric. And the latter links appropriate edges for each vertex with the statistical information of neighbor sequences, thus obtaining an informative and confident graph. Also, we explain UND from the aspect of information theory to verify its validity. Next, with the graph as input, a novel confidence GCN improves confidence edges further to achieve satisfactory clustering results. Extensive experiments demonstrate the effectiveness of our method, which achieves state-of-the-art on face clustering, ReID, and comparable results on fashion datasets. Moreover, because our method has considerable time complexity and stable performance, it can be used for large-scale data cluster cleaning.

Acknowledgments

This work was supported by NSFC 61932002, NSFC 62273347 and NSFC 62272448.

References

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and M. Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:613, 2009. 5

[2] Yingjie Chen, Huasong Zhong, Chong Chen, Chen Shen, Jianqiang Huang, Tao Wang, Yun Liang, and Qianru Sun. On mitigating hard clusters for face clustering. In *European Conference on Computer Vision*, 2022. 3, 5, 6

[3] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13:21–27, 1967. 1

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 1, 3, 5, 6

[5] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society*, 18 5-6:602–10, 2005. 4

[6] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. 4

[7] Senhui Guo, Jing Xu, Dapeng Chen, Chao Zhang, Xiaogang Wang, and Rui Zhao. Density-aware feature embedding for face clustering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6697–6705, 2020. 1, 3

[8] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 5

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[10] Christopher G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 4

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. 1

[12] Junfu Liu, Di Qiu, Pengfei Yan, and Xiaolin Wei. Learn to cluster faces via pairwise classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3825–3833, 2021. 3

[13] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1096–1104, 2016. 5

[14] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136, 1982. 1, 3, 5, 6

[15] Federico Monti, Michael M. Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*, 2017. 3

[16] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004. 5

[17] Mark E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103 23:8577–82, 2006. 5

[18] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344:1492 – 1496, 2014. 3

[19] Martin Rosvall, Daniel Axelsson, and Carl T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178:13–23, 2009. 2, 5

[20] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997. 4

[21] Shuai Shen, Wanhua Li, Zheng Zhu, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou. Structure-aware face clustering on a large-scale graph with 107 nodes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9081–9090, 2021. 1, 3, 5, 6

[22] Yichun Shi, Charles Otto, and Anil K. Jain. Face clustering: Representation and pairwise constraints. *IEEE Transactions on Information Forensics and Security*, 13:1626–1640, 2018. 5

[23] Robin Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16:30–34, 1973. 5, 6

[24] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014. 1

[25] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 1

[26] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 3

[27] Yaohua Wang, Yaobin Zhang, Fangyi Zhang, Senzhang Wang, Ming Lin, YuQi Zhang, and Xiuyu Sun. Ada-NETS: Face clustering via adaptive neighbour discovery in the structure space. In *International Conference on Learning Representations*, 2022. 1, 3, 4, 5, 6, 7

[28] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1117–1125, 2019. 1, 3, 5, 6

[29] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. 5

[30] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016. 1, 3, 5

[31] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6861–6871, 2019. 3

[32] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13366–13375, 2020. 1, 3, 5, 6

[33] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2293–2301, 2019. 1, 3

[34] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. 3

[35] Xiaotian Yu, Yifan Yang, Aibo Wang, Ling Xing, Hanling Yi, Guangming Lu, and Xiaoyu Wang. Facemap: Towards unsupervised face clustering via map equation. *ArXiv*, abs/2203.10090, 2022. 1, 3, 5, 6

[36] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and D. Y. Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *ArXiv*, abs/1803.07294, 2018. 3

[37] Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007. 5