Local Connectivity-Based Density Estimation for Face Clustering

Junho Shin, Hyo-Jun Lee, Hyunseop Kim, Jong-Hyeon Baek, Daehyun Kim, Yeong Jun Koh*
Chungnam National University
{wnsgh8190, gywns6287, hyunseop95, whdgusdl97, seven776484}@gmail.com, yjkoh@cnu.ac.kr

Abstract

Recent graph-based face clustering methods predict the connectivity of enormous edges, including false positive edges that link nodes with different classes. However, those false positive edges, which connect negative node pairs, have the risk of integration of different clusters when their connectivity is incorrectly estimated. This paper proposes a novel face clustering method to address this problem. The proposed clustering method employs density-based clustering, which maintains edges that have higher density. For this purpose, we propose a reliable density estimation algorithm based on local connectivity between K nearest neighbors (KNN). We effectively exclude negative pairs from the KNN graph based on the reliable density while maintaining sufficient positive pairs. Furthermore, we develop a pairwise connectivity estimation network to predict the connectivity of the selected edges. Experimental results demonstrate that the proposed clustering method significantly outperforms the state-of-the-art clustering methods on large-scale face clustering datasets and fashion image clustering datasets. Our code is available at https://github.com/illian01/LCE-PCENet

1. Introduction

Recently, with the release of large labeled face image datasets [7, 9, 10], there are great progresses of face recognition [4, 13, 14, 22]. These data-driven approaches still demand massive annotated face data for improving face recognition models. Face clustering, which aims to divide enormous face images into different clusters, is essential to reduce annotation costs. Also, face clustering can be used in real-world applications, including photo management and organization of large-scale face images in social media, as well as data collection.

Traditional clustering methods such as K-Means [16] and DBSCAN [5] do not require training steps, but they depend on specific conditions on data distributions and are sensitive to hyper-parameters. Also, they work well on small scale-data but are vulnerable to large-scale data. Thus, they are not effective to face image data, which contains large-scale images and diverse distributions in general. Recent researches [2,6,12,17,19,24,26] construct the KNN graph and estimate the connectivity between nodes using deep neural networks with supervised-learning, where the connectivity represents the probability whether two nodes belong to the same cluster. For instance, in [19, 24], graph convolution networks (GCNs) are employed to estimate the connectivity between nodes linked with edges in the KNN graph. Also, the transformer [21] is adopted to exploit the relationship among K nearest neighbors and estimate the connectivity between neighbor nodes [17]. Thus, these methods [17,19,24] estimate the connectivity between most edges in the KNN graph, including many false positive edges, which link nodes with different classes (negative pairs). However, clustering performance is significantly degraded when the connectivity between negative pairs is incorrectly estimated.

Some methods [2, 12, 26] select a small number of edges from the KNN graph to exclude negative pairs and perform classification on the selected pairs only. A GCN-based confidence estimator is designed to select edges that link nodes with higher confidence [26]. Density-based methods [2, 12] estimate density for each node to choose edges that are directed to cluster centers. They pick only one pair for each node based on the estimated density and determine whether each pair belongs to the same cluster through pairwise classification. They achieve high precision performance by reducing negative pairs for classification candidates, but a small number of classification candidates yield relatively low recall performance.

Due to the imperfection of the pairwise classification, the more negative pairs are selected as classification candidates, the more nodes with different classes are likely to be merged in the clustering process. On the other hand, insufficient positive pairs may degrade recall scores since some nodes that belong to the same class cannot be merged. Thus, it is essential to reduce negative pairs while maintaining suffi-

*Corresponding author

cient positive pairs as the pairwise classification candidates for high clustering performance.

In this paper, we propose a novel density estimation method that explores the local connectivity and similarity between K nearest neighbors. First, we construct a KNN graph, where each face image becomes a node, based on the cosine similarity between nodes. Then, we develop a local connectivity estimation network (LCENet), which takes the features of each node and its K nearest neighbors as an input and provides the local connectivity probability between the pivot node and its K nearest neighbors. We then combine the local connectivity and similarity to estimate the reliable density. We refine the KNN graph based on the node density by selecting edges toward cluster centers to exclude false positive edges from the KNN graph. In the graph refinement, we perform density-based and similarity-based edge selection to reduce negative pairs while increasing positive pairs. Given the reconstructed graph, we develop a pairwise connectivity estimation network (PCENet) based on intra-class and inter-class similarities to determine whether or not two linked nodes belong to the same cluster. Finally, we employ the bread-first search (BFS) to obtain clustering results. Experimental results demonstrate that the proposed clustering method significantly outperforms the state-of-the-art clustering methods on large-scale face clustering dataset [7, 25] and fashion image clustering dataset [15].

To summarize, this work has three main contributions.

- We develop LCENet to compute the reliable node density, which effectively excludes negative pairs from the KNN graph while maintaining sufficient positive pairs.
- We design PCENet based on intra-class and inter-class similarities to effectively determine whether two linked nodes belong to the same cluster.
- The proposed clustering method outperforms the state-of-the-arts significantly on various datasets [7, 15, 25].

2. Related Work

Traditional Unsupervised Clustering: Traditional unsupervised clustering methods such as K-Means [16] and DBSCAN [5] have assumptions about data distribution. K-Means assumes all clusters have sphere shape distributions, while DBSCAN supposes that clusters have similar density. Since these assumptions are not suitable for complex real-world data, hierarchical approaches have been studied to be robust on complex distributions. Lin *et al.* [11] proposed the proximity-aware hierarchical clustering, and Zhu *et al.* [30] proposed the rank-order distance, which can replace the fore-used distance metrics. However, these traditional methods fail to provide satisfactory performance on large-scale data.

Supervised Face Clustering: Recent face clustering methods [2,6,12,17–19,24,27,29] employ supervised manners to achieve the high clustering performance on large-scale face data. Though they are based on supervised manner in training phases, they aim at grouping unknown face data whose identities are unseen during training. Thus, in the inference phases, test data has no prior identity information.

Zhan *et al.* [29] generated and aggregated multi-view information to find node pairs, which have reliable edges, and refined the edges using a multi-layer perceptron classifier. Otto *et al.* [18] designed an efficient framework based on K nearest neighbors to achieve lower computational complexity than exploring entire graph edges. Wang *et al.* [24] extracted sub-graphs, whose center is a pivot node, and estimated linkage-probability between nodes. Then, they found edge connections by applying a dynamic threshold to linkage probability. Shen *et al.* [19] estimated KNN edges using the network, which is trained with a novel structure-preserve subgraph sampling strategy, and refined those graphs based on the node intimacy concept. Nguyen *et al.* [17] also used subgraphs, which contain KNN nodes for a pivot, and predicted the relationship between the pivot and the neighbors. Liu *et al.* [12] sampled a small number of pairs from KNN graph and developed the pairwise classification model to classify the sampled pairs into positive or negative.

To exploit the context of the graph, recent face clustering methods have adopted graph convolution networks (GCNs). Inspired by object detection frameworks, Yang *et al.* [27] developed two GCN models for proposal clusters scoring network and outlier node classifier network. Some methods [19, 24] used multiple layers of graph convolution to predict links on a subgraph or entire graph. Yang *et al.* [26] have two networks, GCN-V and GCN-E, which predict node confidence for each node and classify edges from the candidate set. Guo *et al.* [6] constructed a density-aware graph and used GCNs to exploit the local context of nodes. Also, transformer architectures [21] are employed for face clustering [2, 17] to aggregate features of KNN for each node.

3. Methodology

Figure 1 shows an overview of the proposed clustering method. Given a KNN graph, a local connectivity estimation network (LCENet) produces the local connectivity for each node, and reliable node density is estimated based on the local connectivity. Then, density-based and similarity-based edge selection are performed to exclude negative pairs from the KNN graph while boosting positive pairs. Finally, a pairwise connectivity network (PCENet) determines whether the selected node pair belong to the same cluster.

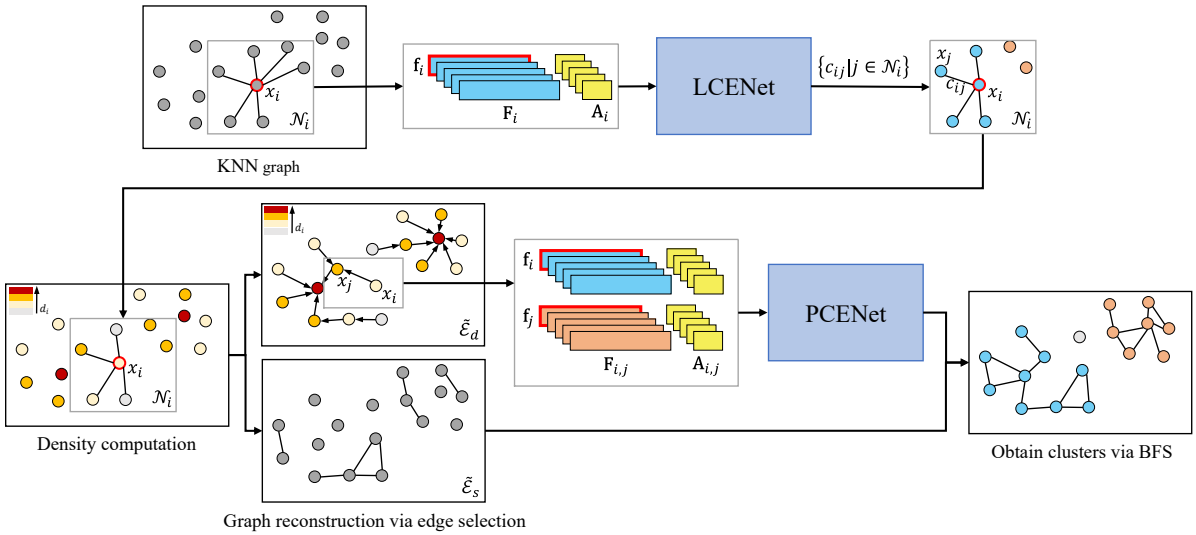


Figure 1. Overview of the proposed clustering method.

3.1. KNN Graph Construction

Given the set of N face images $\mathcal{X} = \{x_i\}_{i=1}^N$, there is the corresponding face feature set $\mathcal{F} = \{\mathbf{f}_i\}_{i=1}^N$, where $\mathbf{f}_i \in \mathbb{R}^D$ is a feature vector with D dimension for x_i, extracted from trained CNNs. Based on the cosine similarity between face features, we form a KNN graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V}$ consists of face image data $\mathcal{X}$ and each node x_i is connected to its K nearest neighbors with edges in the edge set $\mathcal{E}$. Also, let $\mathcal{N}_i$ denote the index set of K nearest neighbors of the node x_i. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be an affinity matrix, in which the (i, j)th element a_{ij} is the affinity between x_i and x_j, which is given by

$$a_{ij} = \begin{cases} \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\|_2 \times \|\mathbf{f}_j\|_2} & j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

3.2. Local Connectivity-based Node Density

Given the KNN graph $\mathcal{G}$, where each node has K edges, it is inefficient to estimate the connectivity of all K pairs for each node since there are KN test pairs. Also, nodes on the boundary among different clusters are connected to neighbors with some false positive edges, where those false positive edges are likely to be assigned incorrect connectivity. For instance, in Figure 2(a), the node x_i has some false positive edges, which link x_i to neighbors that belong to the different cluster, but have similar features to x_i. If one of those false positive edges is assigned the incorrect connectivity by a classification model, two clusters are integrated, which degrades the clustering performance significantly. To address this problem, we select optimal pairs for each node and perform pairwise connectivity estimation to determine whether the selected pairs belong to the same cluster.

To select the optimal pairs for each node, we adopt the concept of node density [2, 6, 12, 26], which links each node to the neighbor node with higher density, with an assumption that the node and its neighbors with higher density have the high probability of being the same cluster. The existing density-based clustering methods [6, 12] try to pair each node with a node that is relatively centered on the cluster. For this purpose, they compute a sum of similarities between a node and its K nearest neighbors to estimate the node density, which represents the measurement of how much a node is close to the center of the cluster. However, the similarity-dependent approach is vulnerable when K nearest neighbors are on boundary among different clusters as in Figure 2(a). For instance, the node x_j has high similarities with its neighbors, even though they belong to different classes. To this end, as in Figure 2(b), x_j is assigned the high density by the similarity-dependent approach, and thus the pair of x_i and x_j are likely to be selected for the pairwise connectivity estimation. The connectivity estimation between x_i and x_j is a hard negative example since they have similar features, resulting in poor clustering results. Therefore, reliable density estimation is essential to pair each node with a more centered node as in Figure 2(c).

Local Connectivity Estimation Network: We develop the local connectivity estimation network (LCENet) to predict connectivity between each node and its K nearest neighbors. We employ the transformer encoder [21] to effectively explore the relationship between each node x_i and its K nearest neighbors, $\{x_j | j \in \mathcal{N}_i\}$. As in Figure 3(a), the transformer encoder consists of 3 identical layers, each composed of a multi-head attention module and a feed-forward network. For each node x_i, we form a local feature matrix $\mathbf{F}_i$ by stacking $\mathbf{f}_i$ and $\{\mathbf{f}_j | j \in \mathcal{N}_i\}$. Here, $\mathbf{f}_i$

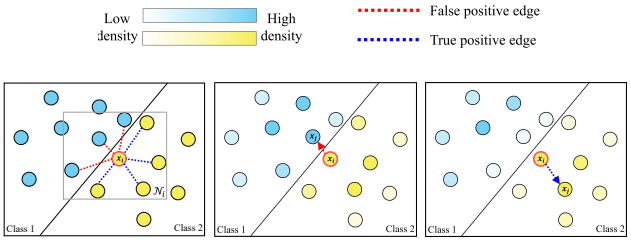


Figure 2. Motivation for reliable density estimator.

is positioned at the first row in $\mathbf{F}_i$. We perform the matrix multiplication to obtain a local affinity matrix $\mathbf{A}_i = \mathbf{F}_i \mathbf{F}_i^T$. We use the concatenation of $\mathbf{F}_i$ and $\mathbf{A}_i$ as the input of the transformer encoder to effectively exploit the relation between neighboring features. The encoder takes $[\mathbf{F}_i\ \mathbf{A}_i]$ and outputs an aggregated feature matrix $\tilde{\mathbf{F}}_i$. We copy the aggregated feature for x_i, which is positioned at the first row in $\tilde{\mathbf{F}}_i$, and concatenate it to each row $\tilde{\mathbf{F}}_i$ along the feature dimension. Then, the concatenated matrix passes through a multi-layer perceptron (MLP) to estimate the local connectivity c_{ij} for all $j \in \mathcal{N}_i$, where c_{ij} denotes a probability that x_i and x_j are in the same cluster.

Training: For training LCENet, we compose the ground-truth of the local connectivity $\{\bar{c}_{ij} | j \in \mathcal{N}_i\}$ for each node x_i. If x_i and x_j belong to the same class, $\bar{c}_{ij} = 1$, otherwise $\bar{c}_{ij} = 0$. We then train LCENet to minimize a binary-cross entropy loss between the estimated connectivity c_{ij} and the ground-truth $\bar{c}_{ij}$. We use all nodes (face images) in the training set to train LCENet.

Density Computation: For reliable density computation, we consider both the local connectivity and the feature similarity between each node x_i and its K nearest neighbors. For each node x_i, we compute a density

$$d_i = \sum_{j \in \mathcal{N}_i} a_{ij} c_{ij}. \tag{2}$$

Unlike the existing methods [6, 12], we use the local connectivity c_{ij} as well as the feature similarity a_{ij} to compute the reliable density. Based on the local connectivity, nodes on the cluster border can be assigned low density, even when they have similar neighbor nodes with different classes.

3.3. Graph Reconstruction via Edge Selection

We reconstruct a graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$, which contains same nodes with $\mathcal{G}$. The reconstructed edge set $\tilde{\mathcal{E}}$ is composed of a density-based edge set $\tilde{\mathcal{E}}_d$ and a similarity-based edge set $\tilde{\mathcal{E}}_s$. The edge set $\tilde{\mathcal{E}}_d$ is determined by selecting a pair for each node based on the node density. For each node x_i, we select neighbor nodes that have higher densities than x_i to construct the set

$$\mathcal{D}_i = \{j | d_j > d_i, j \in \mathcal{N}_i\}. \tag{3}$$

Among nodes $\{x_j | j \in \mathcal{D}_i\}$, we select one node that is most similar to x_i, which is given by

$$j^* = \arg\max_{j \in \mathcal{D}_i} a_{ij} c_{ij}. \tag{4}$$

Then, x_i is connected to x_{j^*} by an edge in $\tilde{\mathcal{E}}_d$. When x_i has no neighbors with higher densities, *i.e.* $\mathcal{D}_i = \emptyset$, x_i is not connected to any neighbor nodes. We perform this process for all nodes in $\mathcal{V}$ to construct the edge set $\tilde{\mathcal{E}}_d$.

Even though we can reduce false positive edges, which connect negative node pairs, in $\tilde{\mathcal{E}}_d$ by selecting at most one edge for each node, it has the limitation that sufficient positive pairs cannot be selected. These insufficient positive pairs may yield over-clustered results, resulting in low recall performance. To address this problem, we additionally form the similarity-based edge set $\tilde{\mathcal{E}}_s$ by connecting x_i to x_j, if $j \in \mathcal{D}_i$ and $a_{ij} c_{ij} > \tau$, where τ is a connecting threshold.

3.4. Pairwise Connectivity Estimation Network

We design the pairwise connectivity estimation network (PCENet) to estimate whether two nodes, connected by edges in $\tilde{\mathcal{E}}$, are in the same class. Figure 3(b) shows the structure of the proposed PCENet. Given a pair of connected nodes x_i and x_j, it yields the pairwise connectivity p_{ij}, which represents the probability that two nodes belong to the same class. To compare two nodes effectively, we use all features of their K nearest neighbors. We form feature matrices $\mathbf{F}_i \in \mathbb{R}^{(K+1) \times D}$ and $\mathbf{F}_j \in \mathbb{R}^{(K+1) \times D}$ using respective K nearest neighbor features as done in LCENet. Notice that $\mathbf{f}_i$ and $\mathbf{f}_j$ are located at the first row in $\mathbf{F}_i$ and $\mathbf{F}_j$, respectively. We concatenate two feature matrices as $\mathbf{F}_{i,j} = [\mathbf{F}_i^T\ \mathbf{F}_j^T]^T \in \mathbb{R}^{(2K+2) \times D}$ and compute an affinity matrix $\mathbf{A}_{i,j} = \mathbf{F}_{i,j} \mathbf{F}_{i,j}^T$, where $\mathbf{A}_{i,j}$ contains both intra-class similarities ($\mathbf{F}_i \mathbf{F}_i^T$ and $\mathbf{F}_j \mathbf{F}_j^T$) and inter-class similarity ($\mathbf{F}_i \mathbf{F}_j^T$). We use $\mathbf{A}_{i,j}$ as well as $\mathbf{F}_{i,j}$ since both intra-class and inter-class similarity scores can reflect whether two nodes are close in embedding space.

Then, $[\mathbf{F}_{i,j}\ \mathbf{A}_{i,j}]$ passes through the transformer encoder, which has the same structure as the transformer encoder in the local connectivity estimation network, to form an aggregated feature matrix $\tilde{\mathbf{F}}_{i,j} \in \mathbb{R}^{(2K+2) \times D}$. Among $2K+2$ aggregated features in $\tilde{\mathbf{F}}_{i,j}$, we pick only two features at the first row and K+2th row, which represent aggregated features $\tilde{\mathbf{f}}_i$ and $\tilde{\mathbf{f}}_j$ for x_i and x_j, respectively. Then, original features and aggregated features are concatenated as $[\mathbf{f}_i\ \mathbf{f}_j\ \tilde{\mathbf{f}}_i\ \tilde{\mathbf{f}}_j]^T$, and the concatenated feature is fed into MLP to analyze the connectivity between nodes x_i and x_j. Thus, the pairwise connectivity estimation network estimates the pairwise relationship p_{ij} between nodes x_i and x_j, representing the probability that two nodes belong to the same cluster.

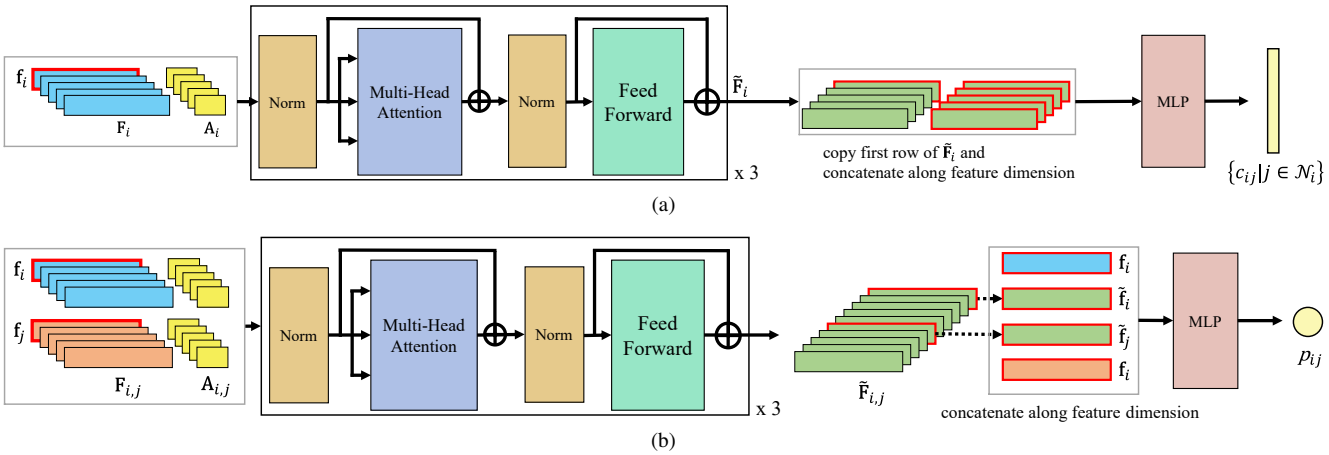


Figure 3. The structures of (a) LCENet and (b) PCENet.

Training and Inference: We refer to pairs with the same class and different classes as positive and negative pairs, respectively. For training PCENet, we collect the ground-truth positive and negative pairs from the training set. For positive pairs, we use all combinations of positive pairs in each class in the training set. For negative pairs, there are too many negative pairs generated when all nodes in the training set are used. Therefore, we form a KNN graph using nodes in the training set and then pick negative pairs connected in the KNN graph. When x_i and x_j are positive pairs, the ground-truth of pairwise connectivity, $\bar{p}_{ij}$ is assigned 1, otherwise $\bar{p}_{ij} = 0$. Then, we train PCENet to minimize a binary-cross entropy loss between p_{ij} and $\bar{p}_{ij}$ for all collected pairs in the KNN graph.

In inference, nodes x_i and x_j are determined to be in the same cluster if $p_{ij} \geq 0.5$. For the efficiency, only pairs connected by edges in $\tilde{\mathcal{E}}_d$ are classified by PCENet, whereas nodes in each pair in $\tilde{\mathcal{E}}_s$ are automatically decided that they share the same cluster. Finally, we employ BFS to obtain clustering results from the sparsely estimated connectivity.

4. Experiments

4.1. Experimental Settings

MS-Celeb-1M: We evaluate the proposed method on the large-scale face clustering benchmark, MS-Celeb-1M [7], which contains 100K identities. Since annotations in the original MS-Celeb-1M is unreliable, we use widely used ones, refined from ArcFace [4], which contains 5.8M face images from 86K identities. We follow the training and test settings in [26], where the dataset is divided into 10 parts with similar sizes. Then, among 10 parts, one part is used for training, while the other parts are used for the test. As done in [2, 12, 17, 19, 23, 26], we evaluate the proposed method on 5 different sizes of test sets from 1, 3, 5, 7, and 9 parts, resulting in 584K (8.57K), 1.74M (25.7K), 2.89M (42.9K), 4.05M (60.0K), and 5.21M (77.2K) images (identities), respectively.

IJB-B: We also evaluate the proposed clustering method on another face clustering benchmark, called IJB-B [25]. As in the setting in the existing face clustering methods [6,12,24], we train LCENet and PCENet using 5K identities and 200K samples in the CASIA [28] dataset, and test the proposed method on three subtasks in IJB-B, referred as $F_{512}, F_{1024}, F_{1845}$. The three subtasks have 512, 1,024, and 1,845 identities with 18,171, 36,474, and 68,195 images, respectively.

DeepFashion: To demonstrate the generalization and the effectiveness of the proposed clustering method, we perform experiments on the DeepFashion [15] dataset, which contains a large number of fashion images. For the fair comparison, we follow the same settings with the existing clustering methods [2, 12, 23, 26], where the training set has 25,752 images from 3,997 categories, and the testing set has 26,960 images from 3,984 categories. The fashion image clustering on DeepFashion is an open-set problem since there is no overlap between training and test categories.

Metrics: For quantitative evaluation, we adopt two popular metrics: Pairwise F-scores and BCubed F-scores [1]. Both F-scores are measured as the harmonic mean of precision and recall. We refer Pairwise F-scores and BCubed F-scores to F_P and F_B, respectively.

Implementation Details: To construct the KNN affinity graph, we use $K = 80$ for MS-Celeb-1M, $K = 120$ for IJB-B, and $K = 8$ for DeepFashion. The connecting threshold τ is computed by the average of similarities between each node and its 3 nearest neighbor for each dataset. The multi-head number is set to 8 for both LCENet and PCENet, and normalization is used before each multi-head attention and feed-forward network. MLP in LCENet and PCENet contains three fully-connected layers. Also, following the existing clustering methods [2, 6, 12, 17, 19, 23, 24, 26, 27], we use pre-trained features for images in MS-Celeb-1M, CASIA, IJB-B, and DeepFashion, provided by [27], [24], [24]

Table 1. Comparison of the proposed method with the existing clustering methods on different number of test images in MS-Celeb-1M. The best results are boldfaced.

Datasets	MS-Celeb-1M									
	584K		1.74M		2.89M		4.05M		5.21M	
Methods/ Metrics	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B
K-Means [16]	79.21	81.23	73.04	75.20	69.83	72.34	67.90	70.57	66.47	69.42
HAC [20]	70.63	70.46	54.40	69.53	11.08	68.62	01.40	67.69	00.37	66.96
DBSCAN [5]	67.93	67.17	63.41	66.53	52.50	66.26	45.24	44.87	44.94	44.74
ARO [18]	13.60	17.00	08.78	12.42	07.30	10.96	06.86	10.50	06.35	10.01
CDP [29]	75.02	78.70	70.75	75.82	69.51	74.58	68.62	73.62	68.06	72.92
L-GCN [24]	78.68	84.37	75.83	81.61	74.29	80.11	73.70	79.33	72.99	78.60
LTC [27]	85.66	85.52	82.41	83.01	80.32	81.10	78.98	79.84	77.87	78.86
GCN(V+E) [26]	87.93	86.09	84.04	82.84	82.10	81.24	80.45	80.09	79.30	79.25
Clusformer [17]	88.20	87.17	84.60	84.05	82.79	82.30	81.03	80.51	79.91	79.95
STAR-FC [19]	91.97	-	88.28	86.26	86.17	84.13	84.70	82.63	83.46	81.47
Pair-Cls [12]	90.67	89.54	86.91	86.25	85.06	84.55	83.51	83.49	82.41	82.40
Ada-NETS [23]	92.79	91.40	89.33	87.98	87.50	86.03	85.40	84.48	83.99	83.28
Chen *et al.* [2]	93.22	92.18	90.51	89.43	89.09	88.00	87.93	86.92	86.94	86.06
Ours	**94.64**	**93.36**	**91.90**	**90.78**	**90.27**	**89.28**	**88.69**	**88.15**	**87.35**	**87.28**

Table 2. Comparison of the proposed method with the existing clustering methods on IJB-B. The best results are boldfaced.

Datasets	IJB-B					
	F_{512}		F_{1024}		F_{1845}	
Methods/Metrics	F_P	F_B	F_P	F_B	F_P	F_B
K-Means [16]	-	61.2	-	60.3	-	60.0
DBSCAN [5]	-	75.3	-	72.5	-	69.5
ARO [18]	-	76.3	-	75.8	-	75.5
L-GCN [24]	-	83.3	-	83.3	-	81.4
DANet [6]	-	83.4	-	83.3	-	82.8
Pair-Cls [12]	84.4	-	83.3	-	82.7	-
Chen *et al.* [2]	80.8	79.6	73.2	78.1	59.1	76.7
Ours	**93.0**	**85.1**	**92.7**	**85.2**	**90.8**	**84.8**

and [26], respectively. We use SGD optimizer with momentum 0.9, learning rate 1e-2, and weight decay 1e-4. We use a single NVIDIA RTX A6000 GPU for both training and inference.

4.2. Comparison with Clustering Methods

We compare the proposed clustering method with various existing clustering methods:

- Traditional clustering: K-Means [16], Density-Based Spatial Clustering Applications with Noise (DBSCAN) [5], Hierarchical Agglomerative Clustering (HAC) [20], and Approximate Rank Order (ARO) [18],

- Learning-based clustering: Consensus-Driven Propagation (CDP) [29], L-GCN [24], Learning to Cluster (LTC) [27], GCN (V+E) [26], Clusformer [17], STAR-FC [19], DANet [6], Pair-Cls [12], Ada-NETS [23], and Chen *et al.* [2].

Table 3. Comparison of the proposed method with the existing clustering methods on DeepFashion. The best results are boldfaced.

Methods	Clusters	F_P	F_B	Time
K-Means [16]	3991	32.86	53.77	573s
HAC [20]	17410	22.54	48.77	112s
DBSCAN [5]	14350	25.07	53.23	2.2s
MeanShift [3]	8435	31.61	56.73	2.2h
Spectral [8]	2504	29.02	46.40	2.1h
ARO [18]	10504	26.03	53.01	6.7s
CDP [29]	6622	28.28	57.83	1.3s
L-GCN [24]	10137	28.85	58.91	23.3s
LTC [27]	9246	29.14	59.11	13.1s
GCN (V+E) [26]	6079	38.47	60.06	18.5s
Pair-Cls [12]	6018	37.67	62.17	**0.6s**
Ada-NETS [23]	-	39.30	61.05	-
Chen *et al.* [2]	8484	40.91	63.61	4.2s
Ours	8842	**41.76**	**64.56**	4.9s

Evaluation on MS-Celeb-1M: Table 1 compares the proposed method with the existing clustering methods on 5 incremental numbers of test images in MS-Celeb-1M. Notice that the proposed method outperforms all existing clustering methods significantly, for example, by margins of 1.42 and 1.18 against the state-of-the-art (Chen *et al.* [2]) in terms of F_P and F_B on 584K test images. It is worth pointing out that the proposed method provides the best scores on large-scale test images, which indicates that the proposed method

Table 4. Ablation study on MS-Celeb-1M, IJB-B (F_{1845}), and DeepFashion for the local connectivity.

Datasets	MS-Celeb-1M 584K		1.74M		2.89M		4.05M		5.21M		IJB-B F_{1845}		DeepFashion	
Methods/ Metrics	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B
Similarity (Sim.)	93.61	91.90	90.52	88.75	88.73	87.03	87.17	85.72	85.85	84.70	89.04	83.74	39.69	64.14
Local Connectivity (LC)	94.12	92.61	91.22	89.69	89.66	88.12	87.88	86.88	86.57	85.90	89.27	84.33	39.39	64.04
Sim. + LC	**94.64**	**93.36**	**91.90**	**90.78**	**90.27**	**89.28**	**88.69**	**88.15**	**87.35**	**87.28**	**90.80**	**84.80**	**41.76**	**64.56**

Table 5. Ablation study on MS-Celeb-1M (584K) and IJB-B (F_{1845}) according to edge selection strategies.

					MS-Celeb-1M (584K)						IJB-B (F_{1845})					
	Sim.	LC	$\tilde{\mathcal{E}}_d$	$\tilde{\mathcal{E}}_s$	BCubed Precision	BCubed Recall	F_B	P.P.	N.P.	$\frac{\text{P.P.}}{\text{P.P.+N.P.}}$	BCubed Precision	BCubed Recall	F_B	P.P.	N.P.	$\frac{\text{P.P.}}{\text{P.P.+N.P.}}$
S_1	✓		✓		97.58	86.38	91.64	554,448	26,282	95.47%	96.14	70.62	81.43	57,618	9,048	86.43%
S_2	✓	✓	✓		96.66	89.66	93.03	558,304	22,032	96.20%	95.78	72.26	82.38	58,074	8,595	87.11%
S_3	✓	✓	✓	✓	96.65	90.28	93.36	1,529,667	22,847	98.53%	95.78	76.10	84.81	529,543	11,297	97.91%

generalizes well to large test sets compared to other methods. The proposed method takes 3.1 minutes using a single GPU to cluster 584K face images in MS-Celeb-1M.

Evaluation on IJB-B: Table 2 shows comparison results of the proposed method with the existing clustering methods on the IJB-B dataset. The scores of the existing methods except [2] are from respective papers. Since Chen *et al.* [2] does not provide the clustering scores on IJB-B, we train their network using the official source code with the default setting and perform the test on IJB-B using the trained network. The proposed achieves the best performance on three test sets for both metrics F_P and F_B. Density-based methods [2, 12] yield lower scores than the proposed method since they pick only one pair for each node for classification, resulting in low recall scores.

Evaluation on DeepFashion: Table 3 provides comparison results of the proposed method with the existing methods on the DeepFashion dataset in terms of F_P, F_B, and the running time to perform the clustering. The scores of the existing algorithms in Table 3 are from [2, 23]. The proposed method provides the best F_P and F_B scores, which indicates that the proposed clustering method has more generalization ability than the other clustering methods. Even though the proposed method is not faster than Pairwise [12], it attains a good trade-off between the performance and running time compared to the existing methods.

4.3. Ablation Study

We conduct the ablation study to validate the efficacy of the proposed LCENet and PCENet.

Efficacy of LCENet: First, we analyze the efficacy of the proposed LCENet. In this paper, we propose the density estimator to select appropriate edges from the KNN graph to exclude negative pairs. As in (2) and (4), both similarity (a_{ij}) and connectivity (c_{ij}) are used to compute the node density and select edges. Table 4 shows the clustering performance on the MS-Celeb-1M, IJB-B (F_{1845}), and DeepFashion datasets, when only similarity is used, only connectivity is used, and both similarity and connectivity are used for density estimation and edge selection. We observe that local connectivity is more effective than similarity for all datasets. Also, the best performance is obtained when both similarity and local connectivity are used. This indicates that the proposed LCENet is essential to estimate the node density and edge selection.

Table 5 compares the numbers of negative pairs (N.P.) and positive pairs (P.P.), which are selected by three edge selection strategies, and their performance on MS-Celeb-1M (584K) and IJB-B (F_{1845}). In the clustering task, negative pairs may significantly degrade the clustering performance when negative pairs are incorrectly determined to be in the same class by the classification. In contrast, insufficient positive pairs may degrade recall scores since some nodes that belong to the same class cannot be merged. Thus, it is essential to reduce negative pairs while increasing positive pairs to improve the clustering accuracy. In Table 5, S_1 uses only similarity (Sim.) to compute the node density and takes the edge set $\tilde{\mathcal{E}}_d$, while S_2 considers both similarity and local connectivity (LC) for the node density. By comparing S_1 and S_2, we observe that S_2, which uses local connectivity obtained from LCENet, effectively reduces negative pairs while boosting positive pairs, resulting in the higher BCubed recall scores than S_1 on MS-Celeb-1M and IJB-B.

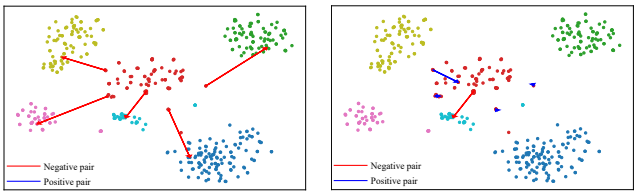
Figure 4 visualizes the embedding space, in which the features of five clusters are depicted as dots in different colors using t-SNE. Negative and positive pairs are depicted by red and blue arrows, respectively. Figures 4(a) and (b) show negative and positive pairs, which are selected by similarity-based density estimation (S_1) and (b) the proposed local connectivity-based density estimation (S_2), re-

Table 6. Comparison of PCENet with LCENet on MS-Celeb-1M, IJB-B (F_{1845}), and DeepFashion.

Datasets	MS-Celeb-1M 584K		1.74M		2.89M		4.05M		5.21M		IJB-B F_{1845}		DeepFashion	
Methods/ Metrics	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B
LCENet	92.05	90.69	89.53	88.38	87.74	87.04	85.90	85.94	84.28	85.07	84.89	79.85	25.91	52.69
PCENet	**94.64**	**93.36**	**91.90**	**90.78**	**90.27**	**89.28**	**88.69**	**88.15**	**87.35**	**87.28**	**90.78**	**84.81**	**41.76**	**64.56**

Table 7. Ablation study on MS-Celeb-1M according to input combinations of PCENet

	Datasets	MS-Celeb-1M 584K		1.74M		2.89M		4.05M		5.21M	
	Combination of Inputs/ Metrics	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B
M_1	$\mathbf{f}_i$, $\mathbf{f}_j$	85.66	84.59	82.88	82.81	80.75	81.63	78.64	80.62	76.57	79.70
M_2	$\mathbf{F}_i$, $\mathbf{F}_j$	83.35	80.85	77.44	77.29	73.95	75.55	71.50	74.26	68.77	73.29
M_3	$\mathbf{F}_i$, $\mathbf{F}_j$, $\mathbf{A}_i$, $\mathbf{A}_j$	93.43	91.88	90.44	88.70	88.69	86.99	87.34	85.68	86.06	84.67
M_4	$\mathbf{F}_i$, $\mathbf{F}_j$, $\mathbf{F}_i\mathbf{F}_j$	93.71	92.53	90.68	89.51	88.96	87.75	87.51	86.35	86.19	85.26
M_5	$\mathbf{F}_i$, $\mathbf{F}_j$, $\mathbf{A}_{i,j}$ (Ours)	**94.64**	**93.36**	**91.90**	**90.78**	**90.27**	**89.28**	**88.69**	**88.15**	**87.35**	**87.28**



(a) (b)

Figure 4. Visualization of negative and positive pairs via (a) similarity-based density estimation (S_1) and (b) the proposed local connectivity-based density estimation (S_2). Negative and positive are depicted by red and blue arrows.

spectively. We see that five nodes on the cluster boundary are connected to nodes with different classes in Figure 4(a), whereas only one negative edge is selected in Figure 4(b). Based on the proposed local connectivity-based density, even the nodes on the cluster border are faithfully linked to positive nodes.

Also, in Table 5, S_3 denotes the proposed clustering method, which uses local connectivity-based density and both edge sets $\tilde{\mathcal{E}}_d$ and $\tilde{\mathcal{E}}_s$. Compared to S_2, S_3 additionally takes $\tilde{\mathcal{E}}_s$ to boost positive pairs for improving recall scores. Through $\tilde{\mathcal{E}}_s$, the proposed method picks more positive pairs while selecting a few negative pairs. To this end, S_3 yields higher recall scores than S_2, especially by a margin of 3.84 on IJB-B.

Efficacy of PCENet: Table 6 shows clustering results on three datasets when PCENet or LCENet is employed to determine pairwise connectivity of edges in $\tilde{\mathcal{E}}_d$. We observe that PCENet provides more accurate clustering results as compared with LCENet. Thus, for reliable clustering results, PCENet is essential, even though LCENet can also provide pairwise connectivity between neighboring nodes.

To compare two nodes x_i and x_j, the proposed PCENet uses all K nearest neighbors of two nodes ($\mathbf{F}_i$ and $\mathbf{F}_j$) and both intra-class and inter-class similarities ($\mathbf{A}_{i,j}$) as an input. In Table 7, we perform experiments with various input settings. By comparing M_1 and M_2, we observe that using neighbor features without similarity degrades performance. Also, in M_3 and M_4, we use only intra-class and inter-class similarities, respectively. By exploiting both intra-class and inter-class similarities, the proposed PCENet (M_5) yields the best performance on all test sets in MS-Celeb-1M.

5. Conclusion

In this paper, we proposed a novel face clustering method based on LCENet and PCENet. First, LCENet provides the local connectivity between neighboring nodes, which is used for reliable node density computation. Second, the density-based edge set is constructed based on the reliable node density. Also, we constructed the similarity-based edge set to obtain sufficient positive pairs for improving recall scores. PCENet predicts the pairwise connectivity possibility of the selected pairs to determine whether each pair belong to the same cluster or not. Extensive experiments demonstrated that the proposed method yields state-of-the-art clustering performance on face clustering and fashion clustering datasets.

Acknowledgement

This work was supported partly by the National Research Foundation of Korea (NRF) grants (NRF-2021R1A4A1031864, NRF-2022R1I1A3069113) and partly by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)).

References

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009. 5

[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 *ECCV*, pages 529–544, 2022. 1, 2, 3, 5, 6, 7

[3] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995. 6

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 1, 5

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 1, 2, 6

[6] Senhui Guo, Jing Xu, Dapeng Chen, Chao Zhang, Xiaogang Wang, and Rui Zhao. Density-aware feature embedding for face clustering. In *CVPR*, pages 6698–6706, 2020. 1, 2, 3, 4, 5, 6

[7] 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*, pages 87–102, 2016. 1, 2, 5

[8] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, volume 1, pages I–I, 2003. 6

[9] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, pages 4873–4882, 2016. 1

[10] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR*, pages 1931–1939, 2015. 1

[11] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 294–301, 2017. 2

[12] Junfu Liu, Di Qiu, Pengfei Yan, and Xiaolin Wei. Learn to cluster faces via pairwise classification. In *ICCV*, pages 3845–3853, 2021. 1, 2, 3, 4, 5, 6, 7

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017. 1

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016. 1

[15] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, pages 1096–1104, 2016. 2, 5

[16] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 1, 2, 6

[17] Xuan-Bac Nguyen, Duc Toan Bui, Chi Nhan Duong, Tien D Bui, and Khoa Luu. Clusformer: A transformer based clustering approach to unsupervised large-scale face and visual landmark recognition. In *CVPR*, pages 10847–10856, 2021. 1, 2, 5, 6

[18] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):289–303, 2017. 2, 6

[19] 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. In *CVPR*, pages 9085–9094, 2021. 1, 2, 5, 6

[20] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973. 6

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 2, 3

[22] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018. 1

[23] 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. 2022. 5, 6, 7

[24] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *CVPR*, pages 1117–1125, 2019. 1, 2, 5, 6

[25] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *CVPRW*, pages 90–98, 2017. 2, 5

[26] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *CVPR*, pages 13369–13378, 2020. 1, 2, 3, 5, 6

[27] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, pages 2298–2306, 2019. 2, 5, 6

[28] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 5

[29] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *ECCV*, pages 568–583, 2018. 2, 6

[30] Chunhui Zhu, Fang Wen, and Jian Sun. A rank-order distance based clustering algorithm for face tagging. In *CVPR*, pages 481–488, 2011. 2