# Learning to Cluster Faces via Confidence and Connectivity Estimation
# Supplementary Material

Lei Yang[1], Dapeng Chen[2], Xiaohang Zhan[1], Rui Zhao[2], Chen Change Loy[3], Dahua Lin[1]
[1]The Chinese University of Hong Kong
[2]SenseTime Group Limited, [3]Nanyang Technological University
{yl016, zx017, dhlin}@ie.cuhk.edu.hk, {chendapeng, zhaorui}@sensetime.com, ccloy@ntu.edu.sg

## 1. Pseudo-code of the proposed algorithm

We provide a pseudo-code to illustrate the steps of the proposed method.

**Algorithm 1** Clustering via Confidence and Connectivity Estimation

**Input:** Graph $\mathcal{G}$, portion of vertices using GCN-E $\rho$, number of connections $M$, cut-off threshold $\tau$
**Output:** Clusters $\mathcal{C}$

Vertex confidence $\mathbf{V}$ = GCN-V($\mathcal{G}$)
$\mathcal{S}$ = GetCandidateSet($\mathbf{V}$)
$\mathcal{H}$ = GetHighConfidenceVertexSet($\mathbf{V}$, $\rho$)
**for** $i \in \mathcal{H}$ **do**
  Edge connectivity $\mathbf{E}_i$ = GCN-E($\mathcal{G}(\mathcal{S}_i)$, $M$)
**end for**
**for** $i \in \mathcal{V} \setminus \mathcal{H}$ **do**
  Edge connectivity $\mathbf{E}_i$ = Max($\mathcal{E}(\mathcal{S}_i)$, $M$)
**end for**
Clusters $\mathcal{C}$ = ConnectToClusters($\mathbf{E}$, $\tau$)
**return** $\mathcal{C}$

## 2. Detailed settings of compared methods

**(1) K-means [4],** minimizes the total intra-cluster variance with a given number of clusters. For $N = 584K$ of MS-Celebe-1M or DeepFashion, we employ K-means by adopting the *ground-truth* number of clusters. For $N \geq 1.74M$, we use mini-batch K-means with batch size $1,000$.
**(2) HAC [6],** adopts *single* strategy for bottom-up merging in our experiments. The distance threshold is set to $0.72$ for different scale of MS-Celeb-1M. For DeepFashion, we tune the distance threshold from $0.1$ to $0.9$ with a step $0.1$ and find $0.4$ gives the best result.
**(3) DBSCAN [3],** has two important hyper-parameters, namely, *radius* and *minPts*. For higher efficiency, we apply $K$NN DBSCAN, which only considers its $K$ nearest neighbors for density computation. We set $K = 80, radius = 0.25, minPts = 1$ for $584K$, $1.74M$ and $2.89M$ of MS-Celeb-1M. When the number of unlabeled images is larger than 4.05M, we have to decrease the distance threshold $\tau$ from $0.25$ to $0.2$, otherwise the pairwise precision will go down to $1.46\%$. For DeepFashion, we set $K = 4, radius = 0.1, minPts = 2$.
**(4) MeanShift [2],** fails to yield results in a reasonable time even on $584K$ of MS-Celeb-1M. Therefore, we only apply the approach in DeepFashion. We tune the *bandwidth* from $0.1$ to $0.9$ and find $0.5$ gives the best result.
**(5) Spectral [5],** has $N \times N$ space complexity, incurring excessive memory demands even on the smallest setting of MS-Celeb-1M ($584K$). We employ spectral clustering on DeepFashion by setting the number of clusters to $3,991$, which is the ground-truth number of clusters.
**(6) ARO [1],** depends on the number of nearest neighbors $K$. For the reported results of MS-Celeb-1M, we use $K = 80$ for all scales. When increasing $K$ to 500, it takes $21h$ to yield $F_P = 54.47$ on $584K$ of MS-Celeb-1M. For DeepFashion, we vary $K$ from 5 to 30 and the best result appears when $K = 10$.
**(7) CDP [9],** adopts a dynamic threshold algorithm to partition the affinity graph efficiently, which relies on an initial threshold $\tau$, a threshold step $\Delta\tau$, maximum size of clusters $s_{max}$ and $K$ for constructing $K$NN affinity graph. For all scales of MS-Celeb-1M, we set $\tau = 0.7$, $\Delta\tau = 0.05$, $s_{max} = 300$ and $K = 80$. For DeepFashion, we set $\tau = 0.5$, $\Delta\tau = 0.05$, $s_{max} = 200$ and $K = 2$.
**(8) L-GCN [7],** adopts the pseudo label propagation algorithm of CDP. In addition to $\tau$, $\Delta\tau$ and $s_{max}$, it requires K at each hop $K_h$ to construct instance pivot graph and active connections $c$ for aggregating the predictions. For $584K$ and $1.74M$ of MS-Celeb-1M, we set $K_0 = 80, K_1 = 10]$, $c = 10$, $\tau = 0.6$, $\Delta\tau = 0.05$ and $s_{max} = 300$. For $N \geq 2.89M$, we increase $\tau$ to 0.7 and $s_{max}$ to 900, while keeping other hyper-parameters the same. For DeepFashion, we set $K_0 = 5, K_1 = 5$, $c = 5$, $\tau = 0.5$, $\Delta\tau = 0.05$ and $s_{max} = 300$.
**(9) LTC [8],** For $N = 584K$ of MS-Celeb-1M, we adopt the same strategy of LTC, which sets different $K$ and $\tau$,

generating a large number of proposals iteratively. For $N \geq 1.74M$, to control the computational budget, we set $K = 80, s_{max} = 300, \Delta\tau = 0.05$ and generate cluster proposals using 5 thresholds ranging from $0.55$ to $0.75$ with a step of $0.05$, without resorting to the iterative scheme. For DeepFashion, we set $K = 5, s_{max} = 100, \tau = [0.55, 0.6]$. Adding proposals generated with $\tau = [0.65, 0.7]$ only increases the $F_P$ from $29.14$ to $29.5$, while increasing the runtime from $13s$ to $27s$.

**(10) Ours (V),** the proposed method mainly relies on two hyper-parameters, namely $K$ and cut off threshold $\tau_c$. For all settings, we set $\tau_c = 0.8$. To construct the $K$NN graph, we set $K = 80$ for MS-Celeb-1M and $K = 5$ for DeepFashion , respectively. For GCN-V, one hidden layer is adopted with a hidden dimension of $512$.

**(11) Ours (V + E),** introduces GCN-E module to select top $\rho$ vertices for connectivity estimation and top-$M$ prediction for connection. For both MS-Celeb-1M and DeepFashion, we set $\rho = 0.7$ for training and $\rho = 0.8$ for inference. $M$ is set to 1 for all settings. To better evaluate the neighborhood of each vertex, we can use different $K$ nearest neighbors for GCN-V and GCN-E. For MS-Celeb-1M, we use $K = 80$ for both GCN-V and GCN-E. For DeepFashion, we use $K = 5$ for GCN-V and $K = 80$ for GCN-E.

## References


[1] Clustering millions of faces by identity. *TPAMI*, 40(2):289–303, 2018. 1

[2] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203. IEEE, 1999. 1

[3] 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*, 1996. 1

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

[5] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002. 1

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

[7] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1117–1125, 2019. 1

[8] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2298–2306, 2019. 1

[9] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *ECCV*, 2018. 1