Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A Graph-based Image Clustering Method Using Mutual Information Maximization

Xun Yu[1], Zicheng Pan[1,2] and Yongsheng Gao[1,*]

[1]*Institute for Integrated and Intelligent Systems, Griffith University*
[2]*School of Information Technology and Electrical Engineering, University of Queensland*
Brisbane, Queensland 4111 Australia

{xun.yu, z.pan, yongsheng.gao}@griffith.edu.au

***Abstract* - In this paper, we present a novel image clustering method that converts a typical image clustering problem into a graph node classification task, given only unlabelled data samples. Graph convolutional network is utilized to perform graph representation encoding. Unlike traditional clustering approaches relying on hand-crafted criteria, our method learns to cluster graph-structured data by maximizing mutual information between the global-graph representation and local-graph representations. The learnt graph embeddings can preserve global information in all locations of local-graphs such as pseudo labels for image clustering tasks. To test the generalization and robustness of the proposed method, we conduct experiments on two different benchmarks, such as fashion classification and fine-grained object classification. The preliminary experiment results show that the proposed method outperforms all the baselines.**

Index Terms - Clustering, graph convolutional network, mutual information maximization, unsupervised learning.

I. Introduction

Thanks to the advances in deep learning technologies, we have witnessed a notable boosting of image classification performance in the recent years [1]. However, most of the supervised deep learning approaches rely on large-scale annotated data samples which would introduce annotation cost and restrict those approaches in limited scenarios. Comparing to supervised learning methods, unsupervised clustering methods can exploit unlabelled data and group them into pseudo classes. However, conventional clustering methods [2-4] rely on impractical assumptions and thus lack the capability to harness complicated real-world data structure. For example, K-Means [2] explicitly assumes that the clusters to be convex-shaped; spectral clustering [4] needs a balanced number of instances in each cluster. Hence, a clustering method without any simplistic assumptions would meet various application scenarios and achieve better clustering performance.

In this paper, we propose an unsupervised learning approach that learns how to cluster from data without any simple assumption limitations. Leveraging the powerful graph convolutional network, we partition the unlabelled data into clusters by maximizing mutual information between the global-graph representation and local-graph representations. Experiments on Fashion-MNIST [5] and Oxford Flowers [6] datasets indicate that our approach can achieve superior performance than all the baselines.

The rest of this paper is organized as follows: Section 2 briefly reviews the related works. Section 3 describes the proposed approach and experimental results are presented in Section 4. In Section 5, we give the work conclusions.

II. Related Work

Clustering methods. As a fundamental problem in many data-driven application domains, data clustering has been extensively investigated and various clustering algorithms have been proposed. Basically, conventional clustering methods can be categorized into several classes, including density based methods [7], centroid based methods [8], connectivity based methods [9], subspace cluster methods [10], hierarchical methods [11] and deep learning methods [12]. Recent papers [13, 14] have used graph convolutional network (GCN) to solve large-scale face clustering and showed notable performance improvement, which demonstrates that GCN can be a powerful tool to solve clustering problem. For a comprehensive review of clustering methods, readers can refer to [15].

Graph convolutional network (GCN). Recently, considerable research effort has been devoted to extending existing deep learning approaches from Euclidean domains to non-Euclidean domains [16]. Typically, GCN methods can be divided into two categories, spectral methods and spatial methods. Spectral approaches [17, 18] work with a spectral representation of the graphs. For example, inspired by the first order graph Laplacian methods, [19] proposes graph convolutional networks (GCNs), which have shown strong capability of modelling complex graphical patterns. Spatial approaches [20, 21], on the other hand, define convolutions directly on the graph, operating on spatially close neighbours. GraphSAGE [22], a inductive framework leverages node features to generate node embeddings for unseen data. LGCN [23], a spatial-GCN method uses fixed number of neighbouring nodes and a sub-graph training method for large-scale graph learning. In terms of learning settings, GCNs can tackle problems in both the transductive settings and inductive settings. In this paper, we propose a graph-based image clustering method which performs node classification in the transductive setting.

Mutual Information. In unsupervised learning, approaches based on mutual information (MI) have been applied to various tasks. MI, as a measure of independence between random variables, plays an important role in independent component analysis (ICA) [24]. However, it is often hard to use MI with deep networks until [25] proposes the Mutual Information Neural Estimation (MINE) that can be used

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

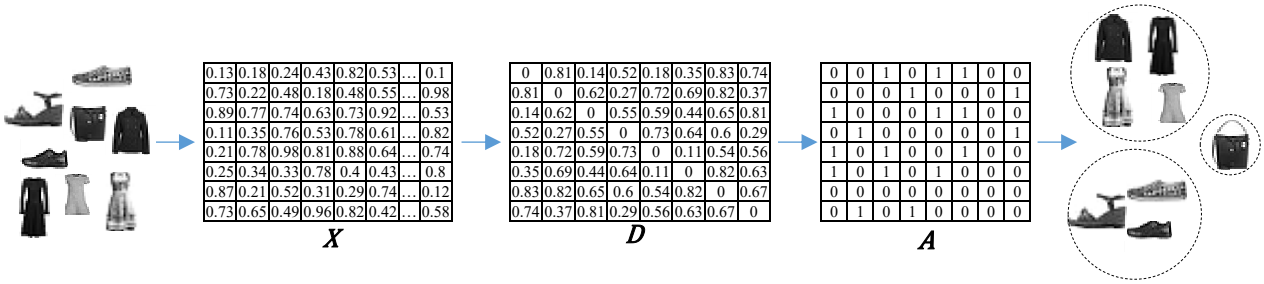

Fig. 1 A toy example to illustrate the process of graph construction. Firstly, the original image set (on the left) will be mapped to a feature matrix $\boldsymbol{X}$. Secondly, we calculate the pairwise distance (for example cosine distance) between each image in $\boldsymbol{X}$ and get a distance matrix $\boldsymbol{D}$. Thirdly, we apply a threshold $\tau = 0.5$ to binary the values in $\boldsymbol{D}$ and generate an adjacency matrix $\boldsymbol{A}$. Lastly, we construct a graph of images (on the right) based on the connectivity information presented in $\boldsymbol{A}$, where each cycle means connected images in this graph.

to learn bi-directional generative models. Following their work, [26] proposes the Deep InfoMax (DIM) that is based on the Jensen-Shannon divergence (JSD) [27] and can prioritize global and local information. Ref. [28] further extends their approach to GCN to learn graph embedding in an unsupervised manner. In this paper, we follow the work in [28] and propose an image clustering method using mutual information maximization.

III. Method

In this section, we present the proposed image clustering approach based on graph convolutional network. The pipeline of this method can be divided into three parts: a) feature extraction, b) KNN graph generation and c) graph node classification. This is a graph-based unsupervised learning method for image clustering tasks.

A. Problem Statement

For a typical image clustering problem, assume that we are given a set of images $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, where N is the number of images in this set and $\boldsymbol{x}_i \in \mathbb{R}^F$ represents the feature of an image. The objective of image clustering is grouping similar images into the same group and assigning a pseudo label for each image group. In this paper, we convert this image clustering problem into a graph representation learning problem. Without knowing the ground truth labels in the training set, a graph embedding method is implemented to encode the original image features to a latent space that maximize the mutual information between a high-level global representation and local representation of parts. This graph embedding learning method intends to preserve the global information of the graph, like the pseudo class of each image cluster.

B. Feature Extraction and Graph Construction

To generate the graph of a provided image set, we firstly conduct a feature extraction process that maps 2D images into 1D feature vectors. Then, a KNN-based method is implemented to the feature space of the image set to construct an adjacency matrix that represents the topological structure of the graph.

Feature Extraction. In this paper, to demonstrate the generalization and robustness of the proposed graph-based image clustering method, we present two feature extraction approaches, raw image pixels and deep unsupervised features extracted from AlexNet [29], VGG16 [30] and ResNet50 [31] (More details will be given in Section 4). Basically, these feature extraction processes F can be formulated as:

$$\mathrm{F}: I_i \in 2D \longrightarrow \boldsymbol{x}_i \in 1D, i = 1,2,\cdots,N \tag{1}$$

where I_i is an image in 2-D space and $\boldsymbol{x_i}$ is a 1-D feature vector, and N is the number of total images. Then, the image set can be denoted as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$.

KNN-based Graph Construction. Given a collection of images $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, we firstly use the cosine similarity metric to calculate the distance between any two images in $\boldsymbol{X}$ and generate a distance matrix $\boldsymbol{D} \in \mathbb{R}^{N \times N}$. Then, we search K nearest neighbors for each image in $\boldsymbol{D}$ and a threshold value τ is used to determine the connectivity of each image in the corresponding adjacency matrix $\boldsymbol{A}$:

$$\boldsymbol{A} \in \mathbb{R}^{N \times N} \; where \; a_{ij} = \begin{cases} 1, & if \; d_{ij} \in \boldsymbol{D} < \tau \; and \; i \neq j \\ 0, & otherwise \end{cases} \tag{2}$$

finally, we obtain an undirected graph $\mathrm{G} = (\mathbf{X}, \mathbf{A})$ where each node is an image and edges between nodes are determined by the adjacency matrix $\mathbf{A}$. Note that threshold τ is a hyperparameter to be set in the training process. A toy example of the proposed feature extraction and graph construction is illustrated in Fig. 1.

C. Graph Node Classification

To leverage the feature information in $\boldsymbol{X}$ and the topological information in $\boldsymbol{A}$, we apply the powerful graph convolutional network (GCN) [19]. Suppose we are given a graph $G = (\boldsymbol{X}, \boldsymbol{A})$, a k-layer GCN consists of k graphs and each of them constructs embeddings for each node by aggregating the embeddings of the node's neighbors in the graph from the previous layers:

$$\boldsymbol{H}^{(k)} = \mathcal{M}\left(\boldsymbol{A}, \boldsymbol{H}^{(k-1)}; \boldsymbol{W}^{(k)}\right) \tag{3}$$

where $\boldsymbol{H}^{(k)} \in \mathbb{R}^{N \times d}$ are the node embedding computed after k layers of the GCN, $\mathcal{M}$ is the message propagation function, which is determined by the adjacency matrix $\boldsymbol{A}$ and $\boldsymbol{W}^{(k)}$ is a learnable matrix. Note that the initialized node embedding $\boldsymbol{H}^{(0)} = \boldsymbol{X}$, which is the original image features.

In this paper, we follow the work in [19] that implements $\mathcal{M}$ using a combination of linear transformations and ReLU activation functions:

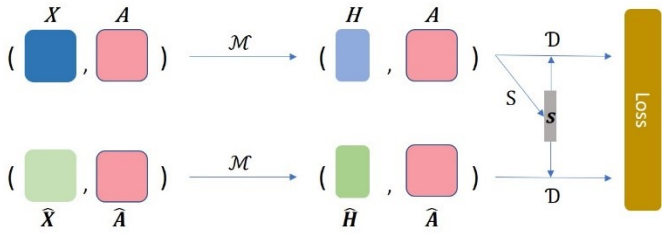

Fig. 2 A brief workflow of the graph node classification process. Best viewed in color.

$$\boldsymbol{H}^{(k)} = ReLU\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k-1)}\right) \tag{4}$$

where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ is the adjacency matrix with inserted self-loops and $\widetilde{\boldsymbol{D}}$ is the corresponding degree matrix $\widetilde{\boldsymbol{D}} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$.

Inspired by [28], we rely on maximizing mutual information between local-graph representations and global-graph representation. Here, the local-graph representation can be a node and its connected neighbors and the corresponding global-graph is the whole graph represented by adjacency matrix $\boldsymbol{A}$. As a contrastive method [32], we design our learning objective by classifying local-global pairs and negative-sampled counterparts. In this way, the learnt graph embeddings can preserve the global information in all locations of local-graphs such as the pseudo labels for each image cluster. The overview pipeline of this unsupervised learning process is described below:

1) Counterparts generation: To preserve the original topological structure of the graph, we keep the adjacency matrix the same $\widehat{\boldsymbol{A}} = \boldsymbol{A}$, but a row-wise permutation operation is applied to the feature matrix $\boldsymbol{X}$ to get counter feature matrix $\widehat{\boldsymbol{X}}$ that can generate different local-graph representations.

2) GCN mapping: Referring to (4), we map $(\boldsymbol{X}, \boldsymbol{A})$ and $(\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{A}})$ to $(\boldsymbol{H}, \boldsymbol{A})$ and $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{A}})$ respectively, where $\boldsymbol{X}, \widehat{\boldsymbol{X}} \in \mathbb{R}^{N\times d_{in}}$, $\boldsymbol{A}, \widehat{\boldsymbol{A}} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{H}, \widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times d_{out}}$. In this paper, we set the length of hidden vector $d_{out} = 512$ and use one-layer GCN in the experiment.

3) Global-graph summarizing & local-graph discriminating: For global-graph summarization, a simple mean aggregation function is implemented to all the node features in $\boldsymbol{H}$ to get global summary score $\boldsymbol{s}$:

$$\boldsymbol{s} = \mathrm{S}(\boldsymbol{H}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{h}_i\right) \tag{5}$$

where σ is the logistic sigmoid function. For local-graph discriminator, we apply a simple bilinear scoring function:

$$\boldsymbol{d} = \mathcal{D}(\boldsymbol{h}_i, \boldsymbol{s}) = \sigma\left(\boldsymbol{h}_i^T\boldsymbol{W}\boldsymbol{s}\right) \tag{6}$$

where $\boldsymbol{W}$ is a learnable matrix and σ is the logistic sigmoid function.

4) Parameter updating: Like the noise-contrastive objective function in [32], a standard binary cross-entropy (BCE) loss function depicted in (7) is utilized to maximize mutual information between $\boldsymbol{s}$ and $\boldsymbol{h_i}$ based on the Jensen-Shannon divergence [27] between the joint and the product of marginals. Using gradient descent, we update the parameters in $\mathcal{M}$ and $\mathcal{D}$.

$$\mathcal{L} = \frac{1}{2N}\left(\begin{array}{l}\sum_{i=1}^{N}\mathcal{M}_{(\boldsymbol{X},\boldsymbol{A})}\left[log\mathcal{D}(\boldsymbol{h_i}, \boldsymbol{s})\right] + \\ \sum_{i=1}^{N}\mathcal{M}_{(\widehat{\boldsymbol{X}},\widehat{\boldsymbol{A}})}\left[log\left(1-\mathcal{D}(\widehat{\boldsymbol{h}_i}, \boldsymbol{s})\right)\right]\end{array}\right) \tag{7}$$

Fig.2 presents the overall workflow of the proposed graph node classification process by maximization mutual information

IV. Experiments

A. Evaluation Metrics and Datasets

Evaluation Metrics. We assess the experimental results by two performance metrics, normalized mutual information (NMI) and pairwise BCubed F-measure [33]. For the first metric, we use Ω and C to represent the ground-truth clusters and the predicted class labels, and define NMI as:

$$NMI(\Omega, \mathrm{C}) = \frac{I(\Omega, \mathrm{C})}{\sqrt{H(\Omega)H(\mathrm{C})}} \tag{8}$$

where $I(\Omega, \mathrm{C})$ is the mutual information and $H(\Omega)$, $H(\mathrm{C})$ are the entropies of Ω and C.

BCubed F-measure, on the other hand, considers both precision and recall. For a data point i, we denote its ground-truth label and cluster label as $L(i)$ and $C(i)$. Then, we define the pairwise correctness between two points i and j as:

$$Correctness(i,j) = \begin{cases}1, if L(i) = L(j)\ and\ C(i) = C(j)\\ 0, \qquad\qquad otherwise\end{cases} \tag{9}$$

The precision P and recall R are defined as:

$$P = Avg_i\left[Avg_{j:C(i)=C(j)}[Correctness(i,j)]\right] \tag{10}$$

$$R = Avg_i\left[Avg_{j:L(i)=L(j)}[Correctness(i,j)]\right] \tag{11}$$

The BCubed F-measure is defined as:

$$F = \frac{2\times P\times R}{P+R} \tag{12}$$

TABLE 1. SUMMARY OF THE DATASETS USED IN OUR EXPERIMENTS.

Dataset	Image type	Image size	Classes	Train/Test Images
Fashion-MNIST	Gray	28×28	10	60,000/10,000
Oxford Flowers	Color	N/A	17	680/680

Fig. 3 Random sampled images from datasets, first row: Oxford Flowers, second row: Fashion-MNIST. Best viewed in color.

TABLE 2. METHOD COMPARISON ON FASHION-MNIST DATASET.

Method	F-Measure			NMI	Time
	Precision	Recall	F-score		
K-Means	0.464	0.452	0.457	0.501	3.03s
SC	0.511	0.524	0.517	0.595	128.12s
HAC	0.446	0.489	0.467	0.526	25.24s
Ours	**0.608**	**0.623**	**0.615**	**0.617**	**0.61s (0.23s)**

TABLE 3. METHOD COMPARISON ON OXFORD FLOWERS DATASET.

Features	Method	F-Measure			NMI
		Precision	Recall	F-score	
AlexNet-conv3	K-Means	0.267	0.278	0.272	0.379
	SC	0.401	0.383	0.392	0.507
	HAC	0.354	0.374	0.364	0.476
	Ours	**0.414**	**0.457**	**0.434**	**0.544**
VGG16-f	K-Means	0.301	0.583	0.397	0.473
	SC	0.357	0.369	0.363	0.479
	HAC	0.392	0.492	0.437	0.537
	Ours	**0.471**	**0.475**	**0.473**	**0.573**
ResNet50-f	K-Means	0.451	0.507	0.451	0.612
	SC	0.417	0.418	0.418	0.544
	HAC	0.533	0.542	0.537	0.621
	Ours	**0.549**	**0.550**	**0.550**	**0.671**

Datasets. In the experiments, we evaluate our approach on two benchmark datasets: Fashion-MNIST [5] and Oxford Flowers [6]. Table 1 lists the details of the above two datasets and randomly selected images are illustrated in Fig. 3. Note that, following transductive learning settings, unlabelled data samples are accessible for training.

B. *Experimental Setup*

Features. To test the generalization and robustness of the proposed method, we implement different feature extractors in the experiments. For the Fashion-MNIST dataset, we simply use raw pixels as feature, which we flatten each 28×28 image to a 784 feature vector. While, for the Oxford Flowers dataset, we implement three different deep unsupervised features that are extracted from AlexNet. VGG16 and ResNet50. To feed the images into different deep network models, we first normalize the image size to 224×224. Then, we use the pre-trained ImageNet models to extract features. More specific, 1) AlexNet-conv3: Following the feature extraction process provided in [34], we add a 4×4 max-pooling layer after the conv3 layer of AlexNet and then get a 3456-dimensional feature for each image. 2) VGG16-f: From the input layer to the last pooling layer of VGG16 is regarded as the feature extraction part of the model, we flatten the output of last pooling layer, resulting in a 25088-dimensional feature vector for each image. 3) ResNet50-f: we follow a similar feature extraction to that of VGG16-f. Without the top fully-connected layers of ResNet50, we get a 100352-dimensional feature vector of per image.

Fig. 4 t-SNE embeddings of the images in the Oxford Flowers dataset from the original Resnet50 features (**left**) and the learned features from the proposed method (**right**). Best viewed in color.

Implementation Details. In the following experiments, we use one-layer graph convolutional network, with hidden feature length $d_{out} = 512$ on Oxford Flowers dataset while $d_{out} = 128$ on Fashion-MNIST dataset due to memory limitations. Adam SGD optimizer is used with an initial learning rate of 0.001and early stopping setting.

C. *Evaluation*

Baselines. For the experiments on Fashion-MNIST dataset, we evaluate the proposed method with three baselines. A brief description of these methods is as follows:1) K-Means [2], the most commonly used clustering method that clusters data by calculating the cluster centroids minimizing the total intra-cluster variance. K-Means requires the number of clusters to be specified. 2) Spectral-clustering (SC) [4], a technique based on graph theory. It makes use of the eigenvalues of the affinity matrix of the data to perform low-dimension embedding, followed by a K-Means in the low dimensional space. 3) Hierarchical agglomerative clustering (HAC) [11], a bottom-up approach that builds nested clusters by merging them successively.

Results. Table 2 shows the results of experiments on Fashion-MNIST dataset. Under F-measure and NMI metrics, we compare with three conventional clustering methods K-Means, spectral-clustering and HAC, and note that the proposed graph-based clustering method outperforms the three baselines under both F-measure and NMI metrics. More importantly, we can find our method show much better performance under F-measure with more than 10% performance improvement compared with other benchmark methods. This suggests that our method is more practical as it still can achieve good performance even considering both precision and recall. In Table 2, we also list the runtime of each method for clustering 10000 test images in Fashion-MNIST dataset. Our approach takes only 0.61s on a CPU and 0.23s under GPU speedup. Even under CPU running environment, our approach is still about 5 times faster than the 2nd fastest K-Means method.

To further test the generalization and robustness of the proposed method, we conduct fine-grained image clustering in Oxford Flowers dataset, which is a more challenging task. Table 3 compares the performance of different methods on this dataset. The results show that, when using stronger deep unsupervised features like ResNet50-f, all the listed clustering methods can gain performance improvement. Under the three different feature extractor settings, our method always outperforms the other baselines, which demonstrates that our

method can benefit from better representations and has superior performance to other baselines. We also provide a t-SNE [35] visualization of the feature embeddings of the original ResNet50-f and the learnt embeddings from our method in Fig. 4.

V. Conclusion

In this paper, we have presented a graph-based image clustering method. We use graph convolutional network to encode graph representation that maximizes the mutual information between local-graph representations and global-graph representation. The learnt graph embeddings can preserve global information in all locations of local-graphs, such as pseudo labels for image clustering tasks. Preliminary experimental results in two datasets demonstrate that the proposed method achieves performance improvement as compared to other baselines. In the future, we aim to test our method in inductive settings and explore some graph training strategies like [36] to overcome the high memory overhead during training stage.

References

[1] P. Druzhkov and V. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis,* vol. 26, no. 1, pp. 9-15, 2016.

[2] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory,* vol. 28, no. 2, pp. 129-137, 1982.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 22, no. 8, pp. 888-905, 2000.

[4] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 11-18.

[5] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747,* 2017.

[6] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1447-1454.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996, vol. 96, no. 34, pp. 226-231.

[8] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 790-799, 1995.

[9] C. Otto, D. Wang, and A. K. Jain, "Clustering millions of faces by identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 40, no. 2, pp. 289-303, 2017.

[10] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245-250.

[11] R. Sibson, "SLINK: an optimally efficient algorithm for the single-link cluster method," *The Computer Journal,* vol. 16, no. 1, pp. 30-34, 1973.

[12] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 3861-3870.

[13] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to Cluster Faces on an Affinity Graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2298-2306.

[14] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1117-1125.

[15] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science,* vol. 2, no. 2, pp. 165-193, 2015.

[16] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596,* 2019.

[17] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203,* 2013.

[18] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163,* 2015.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907,* 2016.

[20] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks,* vol. 20, no. 3, pp. 498-511, 2009.

[21] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2016, pp. 1993-2001.

[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2017, pp. 1024-1034.

[23] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1416-1424.

[24] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski, "Independent component analysis of electroencephalographic data," in *Proceedings of the International Conference on Neural Information Processing Systems*, 1996, pp. 145-151.

[25] M. I. Belghazi *et al.*, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062,* 2018.

[26] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670,* 2018.

[27] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory,* vol. 37, no. 1, pp. 145-151, 1991.

[28] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341,* 2018.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2012, pp. 1097-1105.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[32] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2013, pp. 2265-2273.

[33] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval,* vol. 12, no. 4, pp. 461-486, 2009.

[34] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 132-149.

[35] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, no. Nov, pp. 2579-2605, 2008.

[36] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks," *arXiv preprint arXiv:1905.07953,* 2019.