Image Clustering via Deep Embedded Dimensionality Reduction and Probability-Based Triplet Loss

Yuanjie Yan, Hongyan Hao, Baile Xu, Jian Zhao, *Senior Member, IEEE*, and Furao Shen, *Member, IEEE*

***Abstract*— Image clustering is more challenging than image classification. Without supervised information, current deep learning methods are difficult to be directly applied to image clustering problems. Image clustering needs to deal with three main problems: 1) the curse of dimensionality caused by high-dimensional image data; 2) extracting the effective image features; 3) combining feature extraction, dimensionality reduction and clustering. In this paper, we propose a new clustering framework called Deep Embedded Dimensionality Reduction Clustering (DERC) via Probability-Based Triplet Loss, which effectively solves the above issues. To the best of our knowledge, the DERC is the first framework that effectively combines image embedding, dimensionality reduction, and clustering into the image clustering process. We also propose to incorporate a novel probability-based triplet loss measure to retrain the DERC network as a unified framework. By integrating the reconstruction loss and the probability-based triplet loss, we can improve the image clustering accuracy. Extensive experiments show that our proposed methods outperform state-of-the-art methods on many commonly used datasets.**

***Index Terms*— Image clustering, unsupervised learning, dimensionality reduction.**

I. Introduction

CLUSTERING is one of the most important unsupervised learning methods, which automatically divides a dataset into groups so that the members of each group are similar to each other. Many popular clustering algorithms have been proposed in the literature, including k-means, DBSCAN and spectral clustering [1]. However, when applied directly to high-dimensional data such as images, those algorithms often perform poorly. To improve the performance of image clustering, researchers often turn to image embedding method that reduces the image size and extracts the image features. By mapping high-dimensional images to low-dimensional spaces, image embedding method improves the accuracy of image clustering [2], [3].

Traditionally, various hand-crafted features [4], such as scale invariant feature transform (SIFT) [5] and histogram of oriented gradient (HOG) [6], can be used to obtain effective image invariant features at the cost of high computational complexity. Recently, convolutional neural networks are adopted to extract effective features of images [7], [8]. Among them, multilayer convolutional autoencoders have been used to learn image features on unlabeled images and to compress features for clustering in a unified framework. Those methods can get the image embedding vectors that represent the high-dimensional images in a low-dimensional space.

By utilizing the multilayer convolutional autoencoders, many image clustering methods have been proposed. Deep embedded clustering (DEC) [2] makes use of a deep neural network to learn the image embedded representations and iteratively optimizes the clustering targets using the Kullback-Leibler (KL) divergence loss. The IDEC [9] method improves DEC by combining the reconstruction loss in the autoencoders and the clustering loss. With fully convolutional autoencoders, the discriminatively boosted clustering (DBC) [10] improves the accuracy of image clustering. Deep embedded regularized clustering (DEPICT) [3] consists of a polynomial logistic regression function stacked on top of a multilayer convolutional autoencoder. DeepCluster [11] has a significant advantage in large datasets such as ImageNet [12]. Those network models focus on two parts: the first is to build a suitable image embedding network in which the autoencoder is popularly used in unsupervised learning; the second is to choose the appropriate clustering method, such as k-means and agglomerative clustering. The related methods are summarized by a unified framework in [13].

The methods mentioned above have the following general problems. First, according to the manifold learning hypothesis [14], high-dimensional data can be represented in low-dimensional manifold spaces. However, some clustering methods are not suitable in the low-dimensional manifold space. For example, k-means only works in the Euclidean space. In addition, it is difficult to intuitively obtain the number of specific clusters from the image embedding method, which is crucial for clustering algorithms. Furthermore, some

Manuscript received July 15, 2019; revised January 19, 2020; accepted March 19, 2020. Date of publication April 9, 2020; date of current version April 20, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61876076, in part by the Jiangsu NSF under Grant BK20171344, and in part by the Dengfeng Scholars Program of Nanjing University under Grant B1512002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yannick Berthoumieu. *(Corresponding authors: Jian Zhao; Furao Shen.)*

Yuanjie Yan, Hongyan Hao, Baile Xu, and Furao Shen are with the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210046, China (e-mail: yanyj@smail.nju.edu.cn; haohy@smail.nju.edu.cn; blxu@smail.nju.edu.cn; frshen@nju.edu.cn).

Jian Zhao is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210046, China, and also with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China (e-mail: jianzhao@nju.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2984360

(a) Embedded (b) Reduction (c) Joint train

Fig. 1. Visualization to show the discriminative capability of the embedded subspaces using the MNIST-test dataset. We conduct t-SNE on the image embedded space for Visualization. (a) Clustering in embedded space. (b) Clustering in dimensionality reduction space. (c) Joint training and clustering in the reduced-dimensional space.

inherent defects exist in the embedding space of autoencoder, e.g., Entangled Representations [15], [16]. We also experimentally found that the distribution of embedding vectors belongs to heavy-tailed distributions [17].

To alleviate the above problems, we propose a method that explicitly utilize the dimensionality reduction in this paper. This method obtains sufficient image features through the neural network, and then selects representative features by dimensionality reduction. In our proposed framework, the deep embedded dimensionality reduction clustering (DERC) separates the image clustering processing into three independent steps, i.e., image embedding, dimensionality reduction, and clustering. Various existing methods can be flexibly applied to each part. Furthermore, the DERC framework combines three steps for further optimization. We propose to use the clustering information after dimension reduction to refine the features obtained by the image embedding network. A similar idea is also applied in [18], which utilizes the k-mean to optimize unsupervised linear discrimination analysis (LDA). In this work, we propose a new measure of clustering loss called probability-based triplet loss, which is based on the triplet loss [19] and the Gaussian mixture model (GMM) clustering [20].

An example demonstrating the improvement of our proposed methods is shown in Fig. 1. Fig. 1(a) shows the visualization result of clustering in the initial embedded space before reduction using k-means clustering. Fig. 1(b) shows the clustering results after reducing the dimensions of the image embedding vectors using the t-distributed stochastic neighbor embedding (t-SNE) [21] algorithm. Because t-SNE is also used for visualization, the structures of the two images are the same, but the clustering results are different. Fig. 1(c) visualizes the embedding subspace after retraining the network using the probability-based triplet loss and the reconstruction loss. The main contributions of this paper can be summarized as follows:

- To the best of our knowledge, the DERC is the first framework that separates the image clustering processing into three steps which are image embedding, dimensionality reduction, and clustering. The DERC achieves state-of-the-art in image clustering.
- We propose a new measure of clustering loss called probability-based triplet loss in conjunction with image embedding and clustering for joint training.
- Extensive experiments show that our proposed method outperforms popular clustering methods on some benchmark datasets.

II. Background

In this section, we briefly introduce some techniques related to our framework. The DERC uses the autoencoder network to extract advanced image features, which is called image embedding. Next, we adopt appropriate dimensionality reduction method to filter out useless features and reduce the image embedding vectors from the manifold space to the Euclidean space. Finally, some widespread clustering methods are introduced.

A. Image Embedding

Image embedding uses neural networks to map images to low dimensional vectors which represent images in the embedding space. To extract the effective features, a large number of parameters of the neural network need to be trained on the labeled dataset. It is inconsistent with the purpose of image clustering which is unsupervised learning. Most image clustering models, such as DEC [2], DEPICT [3], take advantage of the autoencoder model [22] which compresses images into low dimensional vectors and then decompress them into high-dimensional vectors that closely matches the original images. Autoencoders are trained to minimize the reconstruction loss, which does not require explicit image labels. Therefore, autoencoders are suitable for image clustering in unsupervised learning. In a sense, autoencoder can also be seen as a dimensionality reduction method. But we are more concerned with the aspect of autoencoders to extract distinguished advanced features from images [23].

B. Dimensionality Reduction

Dimensionality reduction is the process of reducing the number of random variables to mitigate the curse of dimensionality, which has been studied a lot in the literature. There are many methods for dimensionality reduction, such as principal component analysis (PCA) [24], spectral clustering [1] and uniform manifold approximation and projection (UMAP) [25]. There are two main reasons for reducing the size of the image embedding vectors. On the one hand, we get a lot of features in the image embedding process and features may be redundant or similar. Filtering out those useless features is necessary. On the other hand, even the image embedding vectors in low-dimensional manifold space are troublesome to be solved by clustering methods that work on the Euclidean space. We verify this assertion in the experiments.

Although our framework has the flexibility to choose a dimensionality reduction algorithm on a specific clustering task. The t-SNE is better than other reduction algorithms in revealing the structures of datasets by mitigating the congestion problem when mapping from high-dimensional to low-dimensional space. As a result, t-SNE is suitable to process data with the heavy-tailed distribution.

C. Clustering

Clustering consists of a variety of different methods [26]. Depending on the specific datasets, different clustering algorithms have their own strengths and weaknesses. In this paper,

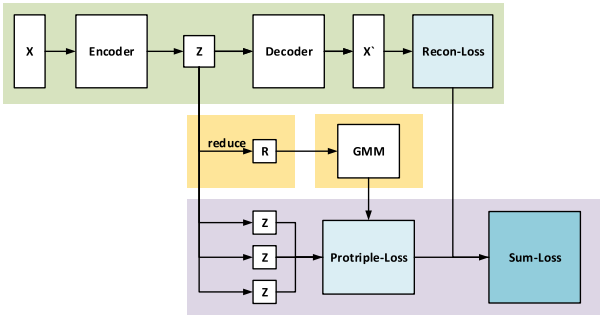


Fig. 2. The DERC framework includes image embedding, dimensionality reduction, and clustering. In addition, these three parts are combined by probability-based triplet loss.

we assume that the image data after dimension reduction is in line with the Euclidean distance, which is verified by many experiments and clustering models. The DEC and the DBC [10] employ the k-means clustering algorithm in the output layer of the encoder. The k-means algorithm can obtain the corresponding cluster centers, and these centers and image embedding vectors are used to calculate the KL divergence loss to retrain the network. The joint unsupervised learning (JULE) [27] is inspired by agglomerative clustering which combines two clusters with the highest affinity in each step until some stopping criteria are met.

The GMM [28] is adopted for clustering in our proposed method. It can not only illustrate the definition of cluster centers but also show the probability of each sample point to the cluster center. We will utilize such probabilities to improve the encoder network by using our proposed probability-based triplet loss.

III. Deep Embedded Dimensionality Reduction Clustering

A. DERC Architecture

Let's consider the task of clustering N images, $X = [x_1, \ldots, x_n]$, into K clusters, where each image $x_i \in \mathbb{R}^{d_x}$ and $d_x = H \times W \times C$. The number of clusters K is usually specified according to the category of the image. As shown in Fig. 2, we divide the DERC network into three parts: image embedding, dimensionality reduction and clustering. In addition, we also proposed a new probability-based triplet loss to retrain the DERC network as a whole. The details of each part of the DERC framework are discussed as follows.

1) Autoencoder Image Embedding: In the image embedding part, we adopt a multilayer convolutional autoencoder. By an encoder network, we can transform x_i to z_i with a nonlinear mapping $f_\theta : X \to Z$, where θ is learned by autoencoder and $Z = [z_1, \ldots, z_n]$ is the image embedded vectors, where each $z_i \in \mathbb{R}^{d_z}$ and d_z represents the image embedding vector's dimension (i.e., $d_z \ll d_x$). The d_z is related to K, which is popular to set $d_z = K$ in many clustering models. However, for complex images with fewer categories, we do not simply set d_z to K. With the help of the decoder function $\varphi_\theta : Z \to X'$ and the reconstruction loss, we initialize the encoder's parameters θ by pre-training the autoencoder. The purpose of image embedding is to extract the high-level features of the image. It is inevitable that some features are redundant or similar between images.

In the autoencoder, the encoder network and the decoder network adopt a symmetrical structure. We only present the structure of the encoder. The encoder network mainly consists of five or six layers of convolutional layers and a fully connected layer at last. The convolutional layer performs downsampling by setting different step sizes to extract the features. The embedding vector z_i is the fully connected layer output from the last layer. Noted that the DERC takes a deeper network than other models. In theory, the ability of the DERC network to extract features is stronger, but the image embedding vectors share some information to hinder the clustering. We use dimensionality reduction to alleviate this problem while exploiting the capabilities of the deep network by the probability-based triplet loss.

2) t-SNE Dimensionality Reduction: To filter out similar features and extract differentiated features, we utilize the t-SNE or other dimensionality reduction algorithms to transform the Z to R, where $R = [r_1, \ldots r_n]$ and $r_i \in \mathbb{R}^{d_r}$ (i.e., $d_r < d_z$). The R is more suitable for clustering than the Z. The basic idea of the t-SNE is that if two data are close in distance in high-dimensional space, they should be close together in the dimensionality reduction space. Mathematically, t-SNE uses conditional probability to describe the similarity between two data. The Gaussian distribution with variance σ_i is constructed centered on point z_i, and the probability that z_j is the neighborhood of z_i is represented by $w_{j|i}$,

$$w_{j|i} = \frac{\exp -||z_i - z_j||^2/(2\sigma_i^2)}{\sum_{k \neq i} \exp -||z_i - z_k||^2/(2\sigma_i^2)}. \tag{1}$$

In the dimensionality reduction space R, using the student's t-distribution with one degree of freedom, $v_{j|i}$ means that the probability of r_j is the neighborhood of r_i, i.e.,

$$v_{j|i} = \frac{(1 + ||r_i - r_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||r_k - r_l||^2)^{-1}}. \tag{2}$$

We learn the nonlinear dimensionality reduction $R = [r_1, \ldots, r_n]$ using the gradient descent method to minimize the KL divergence distance of the W distribution and the V distribution. The t-SNE algorithm is particularly well suited for the dimensionality reduction of high-dimensional datasets and is $\mathcal{O}(n \log(n))$ in time complexity.

3) GMM Clustering: Given the matrix R, we use the GMM to predict the probability of each r_i to clusters. First, we define the probability density function p about sample r on the i-th Gaussian distribution,

$$p(r \mid u_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d_z}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(r-u_i)^T \Sigma^{-1}(r-u_i)} \tag{3}$$

where u_i is the cluster center and Σ_i is the covariance matrix. They are parameters of the i-th Gaussian mixture distribution, and then,

$$p_M(r) = \sum_{i=1}^{K} \beta_i \cdot p(r \mid u_i, \Sigma_i) \tag{4}$$

where $\beta_i > 0$ is a mixture coefficient, $\sum_{i=1}^{K} \beta_i = 1$. The distribution of the dataset is composed of a mixture of K Gaussian distributions. The new parameters β_i, u_i and Σ_i are updated using the EM algorithm [29] as follows,

$$p_{ik} = p_M(y_i = k \mid r_i) = \frac{P(y_i = k) \cdot p_M(r_i \mid y_i = k)}{p_M(r_i)}. \quad (5)$$

Here p_{ik} in (5) indicates the probability of the i-th dimensionality reduced vectors belonging to the k-th cluster. In the last step, we cluster $R = [r_i, \ldots, r_n]$ with a GMM to get the probability of each r_i to different clusters. Please refer to [28] for more details about the GMM method.

B. Probability-Based Triplet Loss

We proposed a novel clustering loss called the probability-based triplet loss to obtain better features of the encoder network, which is inspired by triplet loss [19] and combined with the clustering probability in (5), as follows,

$$D_{ij} = ||z_i - z_j||_2^2 \quad (6)$$

where D_{ij} is the distance between z_i and z_j.

$$L_c(\theta) = \sum_{i}^{n} \sum_{j \in C_a, k \notin C_a} ((1 - ||p_{ia} - p_{ja}||) D_{ij} - D_{ik}) \quad (7)$$

where C_a means the predicted clustering set of data point z_i.

When we retrain the encoder network, we usually add reconstruction loss as regularization [9] to avoid the degradation of the autoencoder. The probability-based triplet loss and the reconstruction loss are combined as follows:

$$L(\theta) = \alpha L_c(\theta) + (1 - \alpha) L_r(\theta) \quad (8)$$

where $L_r(\theta) = \sum_i^n ||x_i - x_i'||^2$ is the reconstruction loss in the autoencoder and $\alpha \in [0, 1]$ is an extra hyperparameter to balance the two losses.

In the previous subsection, we discuss the details of each module independently. However, it would be better to combine them together to handle image clustering problem in [2], [11]. The probability-based triplet loss is used to retrain the autoencoder network by connecting the three parts together. We can optimize the clustering network as a whole to achieve better results than previous clustering. Furthermore, we combine probability-based triplet loss with the reconstruction loss to improve the stability of the encoder network. We observe a significant improvement in performance after using joint training.

The formal definition of probability-based triplet loss is in (7). We intuitively explain the effect of this loss. As shown in the Fig. 3, there is still a situation in which the boundary between the two classes is not clear before retraining. The probability-based triplet loss can minimize the distance in the same cluster set and maintains the intraclass distance as much as possible. At the same time, the distance between different classes is increased.

The probability-based triplet loss can be seen as a variant of the triplet loss combined with the probability of GMM. When the DERC adopts a new clustering method such as DBSCAN, the probability between the pseudo-classes can be set to one, and the probability-based triplet loss will be degraded into a triplet loss.

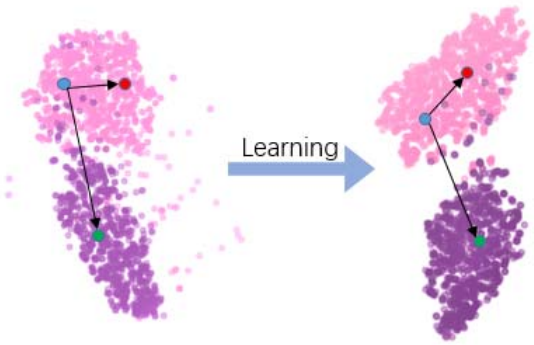


Fig. 3. The experimental results are derived from the t-SNE visualization of the '3' and '5' images in the mnist dataset. The left half is before the retraining and the right half is after the retraining.

C. Optimization

The parameters of the autoencoder network are initialized by Glorot uniform [30]. We optimize those parameters θ using Adam [31] with the default hyper-parameters.

The process of training the DERC network is divided into two phases: pre-training and retraining. In the pre-training process, we first use the reconstruction loss to train the encoder network to obtain the primary image embedding vectors. The encoder's weights θ are updated in mini-batches as follows: $\theta = \theta - \frac{\lambda}{m} \sum_{i=1}^{m} \frac{\partial L_r}{\partial \theta}$, where λ is the learning rate and m is the mini-batch size. Then, we reduce the vectors Z by the t-SNE algorithm to extract distinguishing features and remove redundant features. Meanwhile, the embedding vectors Z are embedded to R from the manifold to a low-dimensional Euclidean space Finally, GMM is used to cluster the reduced dataset R, and the probabilities of each sample to the cluster centers are obtained as P via EM algorithm.

In the retraining process, we refine the autoencoder network by our proposed probability-based triplet loss. The purpose of this step is to optimize the features extracted by the encoder network using the results of clustering. From the previous step, we obtain the pseudo-classes of each dataset C and the corresponding probabilities P. Then, for each retraining sample point x, the DERC randomly selects the same pseudo-class x_s and the different pseudo-class x_d in mini-batches. Through backpropagation, the autoencoder calculate the loss of sample point x to update the weights θ by Eq. (8). In addition, the DERC framework can iteratively retrain the process to improve the performance of clustering. In the discussion section, we will study the relationship between iterative training and clustering accuracy. In summary, Algorithm 1 shows a brief description of the DERC algorithm.

IV. Experiments

In this section, we compare the state-of-the-art clustering methods with the DERC on several benchmark image datasets. By analyzing the experimental clustering results,

Algorithm 1 Deep Embedded Dimensionality Reduction Clustering

Input: images $X = [x_1, ..., x_n]$, the number of clusters K.
Parameter: margin λ, balance α, maximum iterations $iter$.
Output: $C = [C_1, ..., C_K]$, $x_i \in C_j$ and the embedding network parameter θ.

1: Initialize the image embedding network using reconstruct loss.
2: **while** not reach $iter$ **do**
3: Calculate embedding vectors: $Z = f_\theta(X)$ by forward propagation;
4: Compute reduction vectors: $R =$ *t-SNE*(Z);
5: Predict clusters and probabilities: $C, P = GMM(R)$;
6: Generate some triplet data, calculate the $L_c(\theta)$ loss via (7);
7: Retrain network: $\min_\theta L(\theta)$ via (8).
8: **end while**
9: **return** C, θ.

the DERC has an excellent performance compared to other models. Simultaneously, we also carry out ablation experiments to analyze the DERC framework. Through the comparison experiments of each part, we analyze the impact of each part on the DERC algorithm. Our implementations are based on Tensorflow [32] and the code will be published at https://github.com/DizzyDwarf75/DERC. In the end, we experimentally analyze the effects of hyperparameters on the DERC framework.

A. Datasets

The proposed DERC method is evaluated on the following handwritten digit and face image datasets: MNIST-full: a dataset containing 70,000 handwritten digits including 60,000 training samples and 10,000 testing samples, where each sample is a 32 by 32 grayscale image [33]; MNIST-test: a dataset only consisting of the testing samples from MNIST-full; USPS[1]: from the USPS postal service, which contains 11,000 samples of 16 by 16 grayscale image samples; FRGC[2]: using the 20 randomly selected subjects from the original dataset, cropping the face regions and resizing them into 32 by 32 color images; YTF: following [34], we choose the first 41 subjects of YTF dataset. The size of the image is $32 \times 32 \times 3$ pixels; CMU-PIE [35]: a dataset including face images of 68 people with 4 different expressions. The above datasets are also used in the experiments of DEPICT [3]. Table I provides a brief description of datasets.

Note that data preprocessing has a great impact on the training results of the model. Sometimes an effective preprocessing method can lead to significant improvements. To ensure the fairness of comparison with other models, we normalize the images so that the distribution of the data conforms to a standard Gaussian distribution, which is adopted by most models.

[1]http://www.cs.nyu.edu/~roweis/data.html
[2]http://www3.nd.edu/~cvrl/CVRL/Data_Sets.html

TABLE I
DATASETS STATISTICS

Dataset	# Sample	# Classes	Dimensions
MNIST-full	70,000	10	28x28x1
MNIST-test	10,000	10	28x28x1
USPS	11,000	10	16x16x1
FRGC	2,462	20	32x32x3
YTF	10,000	41	55x55x3
CMU-PIE	2,856	68	32x32x1

TABLE II
CONFIGURATION OF THE ENCODER NETWORK

Layer	Configuration
conv1	filters 3x3x2, stride 1x1, pad 0, RELU
conv2	filters 3x3x16, stride 2x2, pad 0, RELU
conv3	filters 3x3x32, stride 1x1, pad 0, RELU
conv4	filters 3x3x128, stride 2x2, pad 0, RELU
conv5	filters 3x3x256, stride 2x2, pad 0, RELU
conv6	filters 3x3x32, stride 1x1, pad 0, RELU
full7	10-128 dimensions

B. Experiment Setup

1) Comparing Methods: We compare the DERC with other clustering algorithms, including k-means, spectral embedded clustering (SEC) [36], agglomerative clustering via path integral (AC-PIC) [37], deep embedded clustering (DEC) [2], deep embedded regularized clustering (DEPICT) [3], joint unsupervised learning (JULE) [27] and discriminatively boosted clustering (DBC) [10]. Moreover, we also do some ablation experiments on the DERC to verify the effectiveness of the dimensionality reduction and probability-based triplet loss. The DAE represents the model only consisting of deep convolutional autoencoder and k-means clustering, which is a typical two-stage model. The model DERC-R differs from the DERC in that it does not use a probability-based triplet loss for retraining.

2) Clustering Metrics: For fairness comparison in different clustering algorithms, we adopt two evaluation metrics, clustering accuracy (ACC) and normalized mutual information (NMI) [38], which are widely used for clustering in unsupervised learning. ACC is measured by the best mapping between the cluster assignments and the true labels using the Hungarian algorithm [39].

3) Implementation Details: We use a deeper autoencoder architecture than any previous models which usually are made of two or there layers of encoder and decoder. The main part of the DERC is an autoencoder which is composed of five or six layers of convolutional networks and a layer of fully connected layer to form an encoder and a symmetric decoder. While for all convolutional layers, the kernel size is 3×3 and the step is 1×1 or 2×2, the full connected layer output is the image embedding vector whose dimension is set to be equal to the number of clusters. But in the experiment we find that setting a larger embedding vector dimension is beneficial for complex images clustering. We believe that the main reason is that high-dimensional vectors can provide richer information. We use the RELU function as the activation function in each layer of convolution and exploit the batch normalization to normalize the output of the fully connected layer. The above contents are summarized in Table II.

TABLE III

CLUSTERING PERFORMANCE ON HANDWRITTEN DATASETS IN TERM OF ACCURACY(ACC) AND NORMALIZED MUTUAL INFORMATION(NMI). THE RESULTS CANNOT BE OBTAINED USING THE MARKS (-)

Dataset	MNIST-full		MNIST-test		USPS	
	NMI	ACC	NMI	ACC	NMI	ACC
k-means	50.0	53.4	50.1	54.7	45.0	46.0
AC-PIC	1.7	11.5	85.3	92.0	84.0	85.5
SEC	77.9	80.4	79.0	81.5	51.1	54.4
DEC	81.6	84.4	82.7	85.9	58.6	61.9
JULE	91.1	96.1	91.1	96.0	91.0	95.0
DBC	91.7	96.4	—	—	74.3	72.4
DEPICT	91.7	96.5	91.5	96.3	92.7	96.4
DAE	78.9	82.4	77.5	80.5	75.8	72.4
DERC-R	91.4	95.6	90.7	94.9	92.3	96.2
DERC	**92.7**	**97.5**	**92.3**	**97.2**	**94.2**	**97.7**

TABLE IV

CLUSTERING PERFORMANCE ON FACIAL DATASETS IN TERM OF ACCURACY(ACC) AND NORMALIZED MUTUAL INFORMATION(NMI). THE ONES MARKED BY (*) ON TOP MEAN THAT THE RESULT COME FROM THE DBSCAN CLUSTERING INSTEAD OF K-MEANS OR GMM. THE RESULTS CANNOT BE OBTAINED USING THE MARKS (-)

Dataset	FRGC		YTF		CMU-PIE	
	NMI	ACC	NMI	ACC	NMI	ACC
k-means	28.7	24.3	77.6	60.1	43.2	22.3
AC-PIC	41.5	32.0	69.7	47.2	90.2	79.7
DEC	50.5	37.8	44.6	37.1	92.4	80.1
JULE-RC	57.4	46.1	84.8	**68.4**	**100.**	**100.**
DEPICT	61.0	47.0	80.2	62.1	97.4	88.3
DAE	54.5	40.9	90.2*	68.0*	73.8	48.1
DERC-R	65.2	49.1	**90.7***	65.8*	94.3	83.7
DERC	**66.7**	**51.3**	—	—	99.6	97.9

It is recommended to make Adam as our optimization method with the default hyper-parameters. The network parameters are initialized by Glorot uniform and the embedding network is pre-trained using reconstruction loss without any other techniques. When we get the embedding vectors from the encoder network, it is necessary to effectively reduce the embedded vectors' dimensionality to obtain differentiated features. After reducing the vectors' dimension with t-SNE, we adopt the GMM to cluster the embedding vectors into image tags.

Finally, we choose valid ternary pairs to calculate the probability-based triple loss to combine the three separate parts. We generate triplet pairs online, which means that when retraining the DERC network, ternary pairs are selected in mini batches. More specifically, a triple is composed of a sample, a positive sample, and a negative sample. Each sample point acts as an anchor, the positive sample has the same pseudo-category as the anchor, and the negative sample has a pseudo-category that is different from the anchor.

C. Comparison With Other Methods

We report the results of all clustering algorithms in Table III about handwritten datasets and in Table IV about facial datasets. We refer to some experimental results of models in [3]. According to the two tables, we can learn that the DERC performs better than the other aforementioned algorithms in general. In terms of accuracy (ACC) and normalized mutual information (NMI), the DERC model can consistently achieve

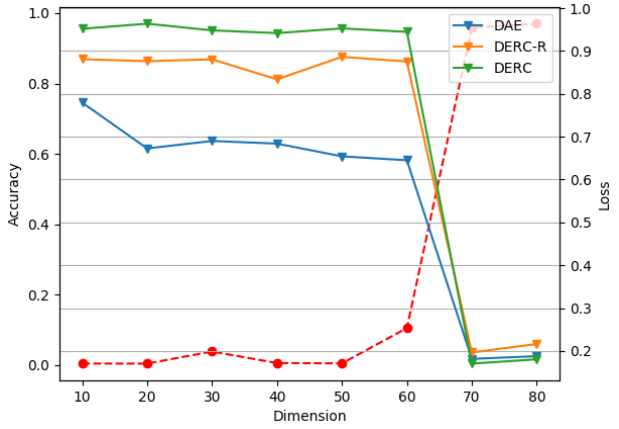


Fig. 4. The clustering accuracy of each model and the image embedding dimensions on MNIST-test. The red line represents the reconstruction loss of the autoencoder.

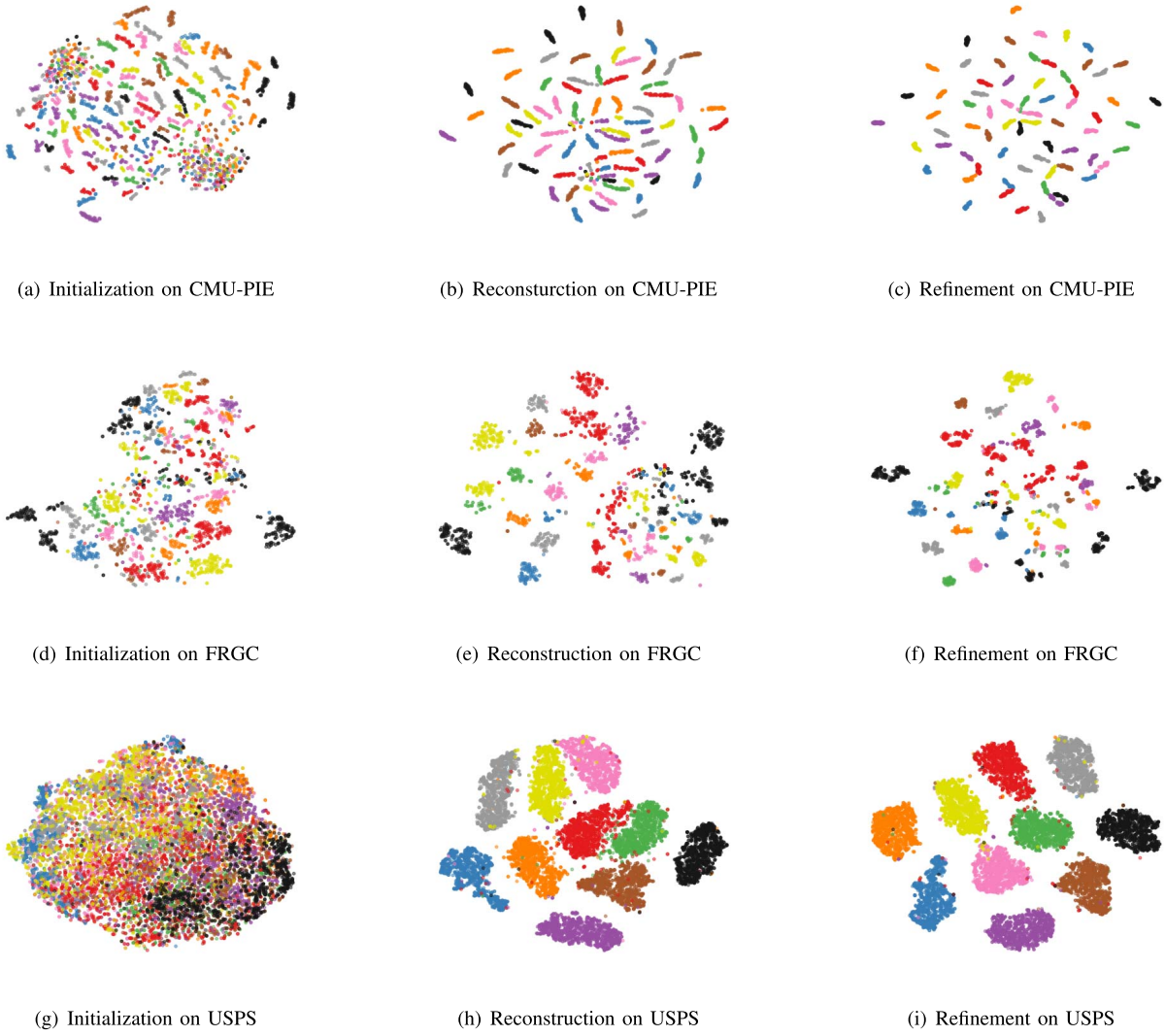
better results on handwritten datasets. On the facial datasets, the DERC also achieved competitive clustering results. Simultaneously, the DERC framework has greater flexibility and adaptability than any other model. From image visualization of reducing image embedding vectors, we decide more intuitively to apply an appropriate clustering method to cluster the data. For examples, on the YTF dataset, we adopt a simple density clustering approach to achieve state-of-the-art.

Please note that the other models, such as JULE and DEC report their best results by tuning hyperparameters, but the DERC does not spend much time tuning and is robust to hyperparameters in some sense. The DEPICT model can be seen as increasing the data by adding random noise to the original image, while our model only performs a simple normalization of the images. Both the JULE and the AC-PIC use agglomerative clustering which is slower than k-means or GMM clustering [3].

D. Comparison With Ablation Models

We set up the DAC without dimensionality reduction and the DERC-R without refining process as comparative experimental models to explore the effectiveness of the various parts of the proposed framework. In Table III and Table IV, compared with the DAE's results, DERC-R has better performance in clustering. This means that the dimensionality reduction process is beneficial for GMM clustering. Moreover, the retrained DERC method is also superior to the DERC-R method. With the probability-based triplet loss, we improve the accuracy of the cluster by approximately 2% after retraining on benchmark datasets. Other methods, such as DEPICT, etc., their framework is similar to the DAE method. The clustering accuracy of the DERC-R framework without retraining process is lower than that of the DERC framework.

In Section V, we also explored the influence of the size of image embedding on the clustering results in Fig. 4. When different methods work on MNIST-test at the same image embedding dimensions, similar results can be obtained. We experimentally verify the effectiveness of the dimensionality reduction and retrain methods. In addition, visual analysis intuitively shows the role of the respective modules in the



(a) Initialization on CMU-PIE (b) Reconsturction on CMU-PIE (c) Refinement on CMU-PIE

(d) Initialization on FRGC (e) Reconstruction on FRGC (f) Refinement on FRGC

(g) Initialization on USPS (h) Reconstruction on USPS (i) Refinement on USPS

Fig. 5. Visualization to show the discriminative capability of embedding subspaces. We conduct t-SNE on the image embedding space for Visualization. (a) Visualize in embedding space when the autoencoder is initial. (b) Visualize when the embedding network is trained using reconstruction loss. (c) Visualize when we utilize the probability-based triplet loss and reconstruct loss to refine the network. Different colors represent the ground-truth of the label. Note that we only use ten colors to represent different classes, so if the number of categories is greater than 10, different classes will have same colors.

next section. In summary, we experimentally verify the effectiveness of the t-SNE dimensionality reduction method and retraining process via the probability-based triplet loss for image clustering.

V. Discussion

We discuss some practical issues and related researches that are faced in applying this framework. First, we explore the hyperparameters setting for image embedding. Next, we show the distribution of embedding vectors and study the effect of iteratively refining image embedding network on clustering accuracy. Then, some clustering results are visualized. Finally, we discuss the failed case and give some effective suggestions.

A. Hyperparameters of Image Embedding

Network structure and image embedding size are critical for the autoencoders. The network structure of the autoencoder adopts the popular CNN structure for processing images [7]. Under the same general network architecture, we study the effect of the dimensions of image embedding on clustering accuracy. From Fig. 4, it can be concluded that DERC is not sensitive to the dimensions of image embedding. Note that the accuracy of the cluster suddenly drops when the dimension is approximately 70. It is due to the inability of the embedding network to converge. We set better hyperparameters to achieve a convergence of the embedding network. The final clustering accuracy reaches the same level as before.

B. The Distribution of Embedding Vectors

We found that the distribution of embedding vectors in the same clustering coincides with heavy-tailed distributions in Fig. 6. We obtain 128-dimensional image embedding vectors on CMU-PIE dataset. and we rank the absolute average value of each dimension in the same clustering

TABLE V

THE ACC AND NMI PERFORMANCE ABOUT BALANCE FACTOR α ON THE MNIST-TEST DATASET

α	0.1	0.3	0.5	0.7	0.9
ACC	96.60	96.76	96.76	98.25	96.71
NMI	91.85	92.12	92.22	93.10	92.06

TABLE VI

THE ACC AND NMI PERFORMANCE FOR DIFFERENT ITERATIONS OF TRAINING ON THE USPS DATASET

Iteration	0	1	2	3	4	5
ACC	95.75	97.37	98.11	98.25	98.80	98.90
NMI	92.33	94.74	95.79	96.10	96.44	96.53

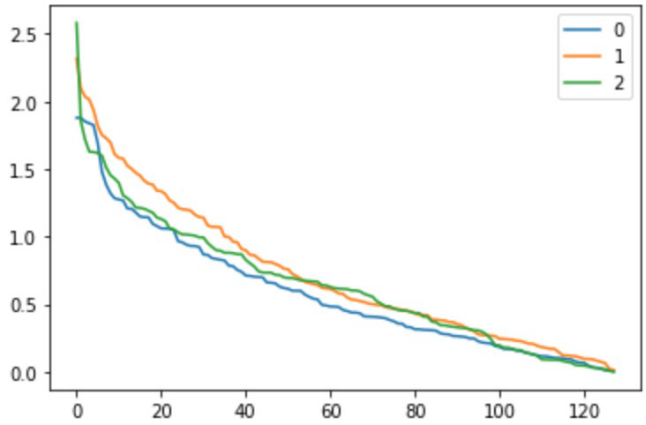


Fig. 6. The distribution of the embedding vectors about three categories.

embedding vectors. Then, we show the distribution of embedding vectors in the same cluster.

Moreover, the embedding space learned by autoencoder can be regarded as the manifold space [40], [41]. However, GMM clustering is engaged in on Euclidean space. The t-SNE dimensionality reduction is suitable to map the embedding vectors to Euclidean space and deal with the information of the heavy-tailed distribution. We think that it is a reason that the t-SNE method can improve the GMM clustering method for image embedding.

C. Hyperparameters of Iterative Training

The probability-based triplet loss takes advantage of the pseudo-class results of clustering after dimension reduction to refine the image embedding network. In the Table. VI, we study the iterative training on the clustering results to verify the effect of the proposed loss. We discover that the clustering result can be significantly improved during the first iteration of retraining. However, as the number of iterations increases, the improvement in accuracy is not significant. In Table. V, we show the effect of α in Eq. (8) on MNIST-test. Balance factor α has good adaptability for clustering. What's more, it can be seen that appropriately increasing the weight of the probability-based triplet loss can improve the clustering method slightly.

D. Visualization of the Image Embedding

In Fig. 5, we visualize the results using image embedding at different stages. The image embedding representations are shown in three stages: 1) initialization stage, where the network parameters are randomly initialized; 2) reconstruction stage, where the network parameters are trained only using reconstruction loss; 3) refinement stage, where the embedding network parameters are retrained using both probability-based triplet loss and reconstruction loss. We visualize the USPS, FRGC and CMU-PIE datasets, which are considered to be representative, including the case of few classes and multiple samples, the case of few classes and few samples, and the case of multiple classes but few samples. From the visual analysis of multiple datasets, we can more intuitively understand the improvement of clustering caused by dimensionality reduction and retraining.

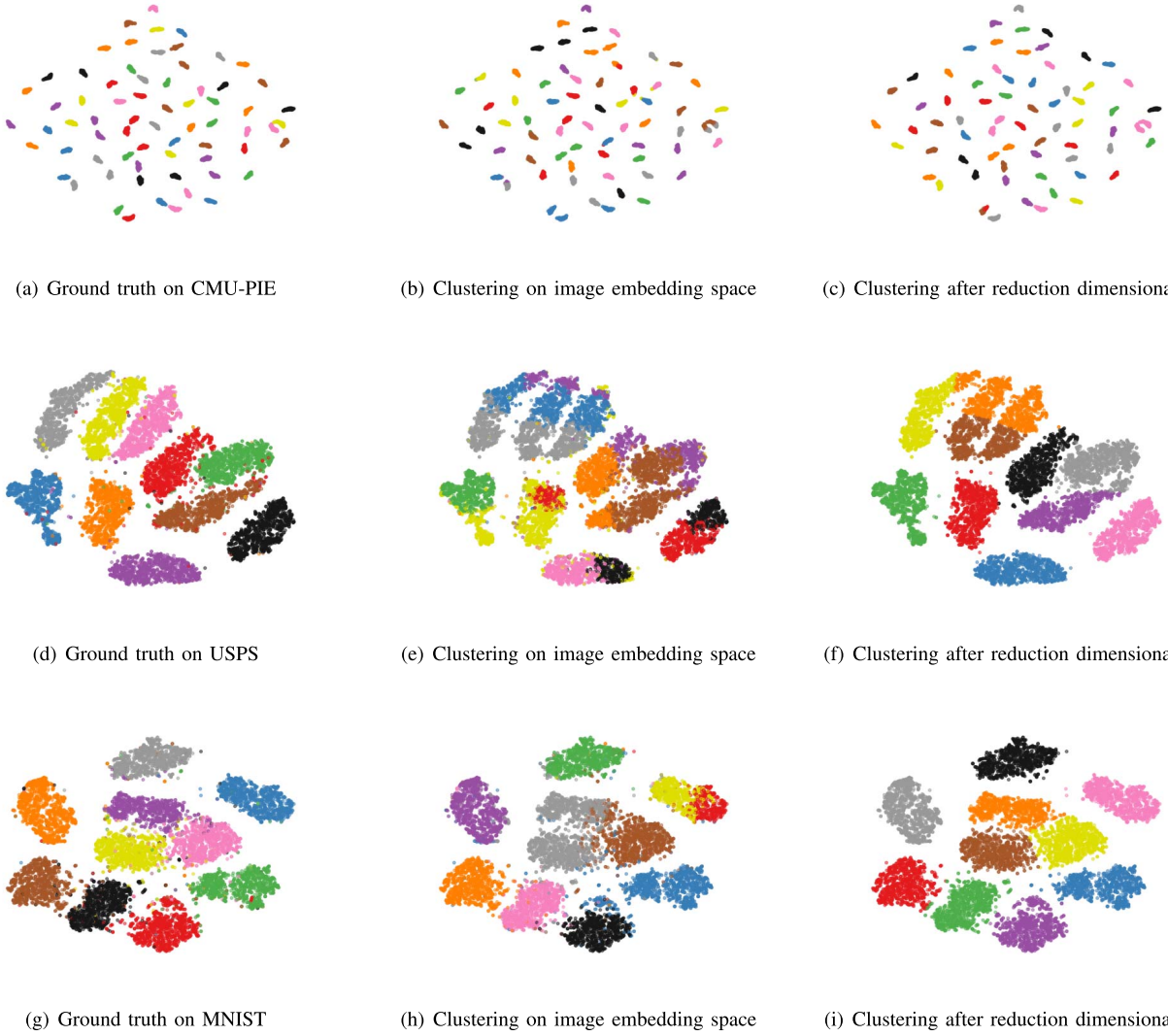
Fig. 7. Clustering samples on the FRGC dataset. Each column is some randomly selected samples of the same cluster.

E. Visualization of the Clustering

In Fig. 5, we explore the impact of clustering directly after embedding, i.e., without the reduction method. The need for dimensionality reduction was verified by ablation experiments. Furthermore, in Fig. 8, we mainly analyse the clustering results of initialization, pre-training and re-training of the DERC to verify the effectiveness of the proposed probability-based triplet loss. Fig. 8 shows the visualization of direct clustering and clustering after dimension reduction on image embedding. The visualization results on multiple datasets show that the performance of general clustering methods such as GMM on the image embedding space is not ideal but improved by the dimension reduction method. Fig. 7 shows the results of clustering samples on the FRGC dataset. It can be seen that the DERC method effectively cluster faces with different illumination and slight deformation.

F. The Failure Clustering

Although the DERC has such excellent results, it still fails to achieve the desired goal for a variety of reasons. There



(a) Ground truth on CMU-PIE (b) Clustering on image embedding space (c) Clustering after reduction dimensionality

(d) Ground truth on USPS (e) Clustering on image embedding space (f) Clustering after reduction dimensionality

(g) Ground truth on MNIST (h) Clustering on image embedding space (i) Clustering after reduction dimensionality

Fig. 8. Visualization to show the performance of clustering on the different periods. We conduct t-SNE on the image embedding space for Visualization. (a) Visualize ground truth in embedding space. (b) Visualize clustering results in the image embedding space using the GMM. (c) Visualize clustering results after reduction dimensionality in the image embedding. Different colors represent the ground-truth of the label. The same color in different subgraphs has no meaning. We only use ten colors to represent different classes, so if the number of categories is greater than 10, different classes will have same colors.

are two main reasons for the failure of the DERC: the image embedding process and the clustering process. From Fig. 3, when the reconstruction loss of the image embedding network is stabilized at a large local minimum, it is disadvantageous for subsequent dimensionality reduction and clustering. Although it is not guaranteed that the image embedding network can converge to a good local minimum point, by monitoring the specific reconstruction loss, we can select different hyperparameters to run the network multiple times to achieve the ideal convergence point.

VI. Conclusion

In this paper, we propose a novel deep embedded dimensionality reduction framework (DERC) for clustering images. The DERC framework consists of three parts: image embedding, dimensionality reduction and clustering. Image embedding primarily handles feature extraction of images. Dimension reduction on image embedding facilitates visual analysis while facilitating the selection of clustering methods so that it improves the clustering accuracy. The choice of clustering methods is also more flexible and different methods can be selected based on the dataset. We also proposed probability-based triplet loss to optimize the image embedding network and use clustering pseudo-class results to achieve the state-of-the-art.

In our future work, we shall improve the DERC framework using some measures, such as focusing on the network to replace the convolutional network and to combine it with semi-supervised clustering.

References

[1] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[2] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[3] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5736–5745.

[4] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 73–86.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Proc. 1999, pp. 1150–1157.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[7] X. Yao, X. Feng, G. Cheng, J. Han, and L. Guo, "Rotation-invariant latent semantic representation learning for object detection in VHR optical remote sensing images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 1382–1385.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[9] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[10] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognit.*, vol. 83, pp. 161–173, Nov. 2018.

[11] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11218, 2018, pp. 139–156.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[13] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," 2018, *arXiv:1801.07648*. [Online]. Available: http://arxiv.org/abs/1801.07648

[14] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *J. Amer. Math. Soc.*, vol. 29, no. 4, pp. 983–1049, Feb. 2016.

[15] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2610–2620.

[16] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," 2018, *arXiv:1811.12359*. [Online]. Available: http://arxiv.org/abs/1811.12359

[17] S. Foss *et al.*, *An Introduction to Heavy-Tailed and Subexponential Distributions*, vol. 6. New York, NY, USA: Springer, 2011.

[18] F. Wang, Q. Wang, F. Nie, Z. Li, W. Yu, and R. Wang, "Unsupervised linear discriminant analysis for jointly clustering and subspace learning," *IEEE Trans. Knowl. Data Eng.*, Sep. 4, 2019, early access, doi: 10.1109/TKDE.2019.2939524.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[20] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, Feb. 1999.

[21] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[22] K. Han *et al.*, "Variational autoencoder: An unsupervised model for encoding and decoding fMRI activity in visual cortex," *NeuroImage*, vol. 198, pp. 125–136, Sep. 2019.

[23] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *Proc. SIGGRAPH ASIA Tech. Briefs (SA)*, 2015, p. 18.

[24] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

[25] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*. [Online]. Available: http://arxiv.org/abs/1802.03426

[26] T. S. Madhulatha, "An overview on clustering methods," *IOSR J. Eng.*, vol. 2, no. 4, pp. 719–725, Apr. 2012.

[27] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5147–5156.

[28] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Cambridge, U.K., Aug. 2004, pp. 28–31.

[29] G. J. McLachlan and T. Krishnan, *The EM Algorithm Extensions*, vol. 382. Hoboken, NJ, USA: Wiley, 2007.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Sardinia, Italy, May 2010, pp. 249–256.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement.*, Savannah, GA, USA, Nov. 2016, pp. 265–283.

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[34] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.

[35] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 53–58.

[36] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and Out-of-Sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[37] W. Zhang, D. Zhao, and X. Wang, "Agglomerative clustering via maximum incremental path integral," *Pattern Recognit.*, vol. 46, no. 11, pp. 3056–3065, Nov. 2013.

[38] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," in *Proc. GSCL*, 2009, pp. 31–40.

[39] M. Jünger *et al.*, *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*. New York, NY, USA: Springer, 2009.

[40] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[41] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, Mar. 2018.