Information Sciences 568 (2021) 199–215

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Learning deep discriminative representations with pseudo supervision for image clustering

Weibo Hu [a,b], Chuan Chen [a,b,*], Fanghua Ye [c], Zibin Zheng [a,b], Yunfei Du [a,b]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[b] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China
[c] Department of Computer Science, University College London, Gower Street, London, UK

ARTICLE INFO

Article history:
Received 29 October 2020
Received in revised form 29 March 2021
Accepted 31 March 2021
Available online 6 April 2021

Keywords:
Image clustering
Deep joint clustering
Pseudo supervision
Pseudo labels
Discriminative representation learning

ABSTRACT

Image clustering is a crucial but challenging task in machine learning and computer vision. Its performance highly depends on the quality of image feature representations. Recently, deep joint clustering which combines representation learning with clustering has presented a promising performance. However, existing joint methods suffer from two severe problems. That is, the learned representations lack discriminability especially for intricate images, and the performance often encounters a bottleneck due to the lack of supervision information. To address these problems, we propose a pseudo-supervised joint method for image clustering, i.e., *Discriminative Pseudo Supervision Clustering* (DPSC). Our key idea is to discover and utilize the pseudo supervision information to provide supervisory guidance for discriminative representation learning. With the aid of pseudo supervision, the representations can be continuously refined to facilitate inter-cluster separability and intra-cluster compactness, thereby leading to more discriminative representations and correctly separated clusters. To fully benefit from joint learning, we further introduce a self-evolution training algorithm to jointly optimize the DPSC model, in which the learned representations and clustering results boost each other progressively as more reliable pseudo supervision information is discovered during the iteration. Experimental results show that DPSC significantly outperforms state-of-the-art methods on various image datasets. Moreover, the learned feature representations generalize well across various algorithms.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Image clustering is a crucial technique in machine learning and computer vision [6], which has boosted various applications such as image segmentation, image annotation, image retrieval, and so on [48,8,10]. However, image clustering is a hard and challenging task since image data are high-dimensional and objects in images often possess local structures. In general, the performance of image clustering relies heavily on the quality of the extracted image features, thus traditional clustering algorithms that directly take the image intensity as input cannot provide satisfactory performance when applied to image clustering [46].

[*] Corresponding author at: School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.

E-mail addresses: huwb7@mail2.sysu.edu.cn (W. Hu), chenchuan@mail.sysu.edu.cn (C. Chen), smartyfh@outlook.com (F. Ye), zhzibin@mail.sysu.edu.cn (Z. Zheng), duyunfei@mail.sysu.edu.cn (Y. Du).

https://doi.org/10.1016/j.ins.2021.03.066
0020-0255/© 2021 Elsevier Inc. All rights reserved.

To obtain better image clustering results, tremendous efforts have been dedicated to learning better image feature representations. Traditionally, various hand-crafted feature descriptors such as SIFT [29] and LBP [19] have been used to encode the images. Recently, numerous deep unsupervised methods have been proposed to learn informative representations [2,39,32]. Technically, these methods aim to encode inputs into low-dimensional representations and decode them to reconstruct the inputs by using a symmetric network structure, i.e., autoencoder. The methods combining representation learning with clustering can be divided into two categories. (1) Sequential methods that firstly learn the representations independently with clustering algorithms following. (2) Joint methods that learn the representations and cluster assignments simultaneously.

Sequential methods are the pioneer to combine representation learning with clustering. Typically, they firstly train the network by minimizing the reconstruction loss to project the raw data into the embedded feature space, and then K-means or other clustering algorithms can be applied as post-processing. The sequential methods have presented an impressive improvements over hand-crafted features [21,38]. However, due to the completely separated process of representation learning and clustering, the learned features may not be reliable for clustering and also cannot be further improved to achieve better performance. To address this problem, extensive joint methods have been proposed to unify the two processes into a single model [43,15,44,12]. Besides, by replacing the fully-connected autoencoder with convolutional autoencoder, the performance of image clustering has been further enhanced [13,16,28,45,22].

In summary, we find that the combination of neural networks and clustering-oriented objectives prevails in the deep clustering problem. Existing joint methods mainly focus on determining a more proper combination, i.e., using a fully-connected or convolutional autoencoder or convolutional neural network to extract image features, and combining with different clustering-oriented objectives such as K-means, soft clustering, or unsupervised cross-entropy to yield image clusters [1].

Despite their huge popularity, existing joint methods still suffer from several severe problems and challenges. (1) They usually cannot learn sufficiently discriminative feature representations, especially for intricate or indistinguishable images, such as faces and real objects, leading to entangled clusters, since they lack explicit cluster-promoting constraints on the latent feature representations during training. (2) The performance encounters a bottleneck due to the lack of supervision information in the joint learning process. (3) It is difficult and costly to specify a combination that performs well across various datasets. Thus, a more general and effective way to improve clustering performance is imperative.

To address these problems, we pave a new way and propose a pseudo-supervised joint clustering method, namely *Discriminative Pseudo Supervision Clustering* (DPSC), which discovers the pseudo labels from the clustering results and exploits them to learn more discriminative representations, thus enhancing the clustering performance. The essential insight behind DPSC is that benign feature representations are beneficial for clustering, and inversely clustering results can also be treated as pseudo labels, therefore providing supervisory guidance for representation learning. The framework of DPSC is shown in Fig. 1.

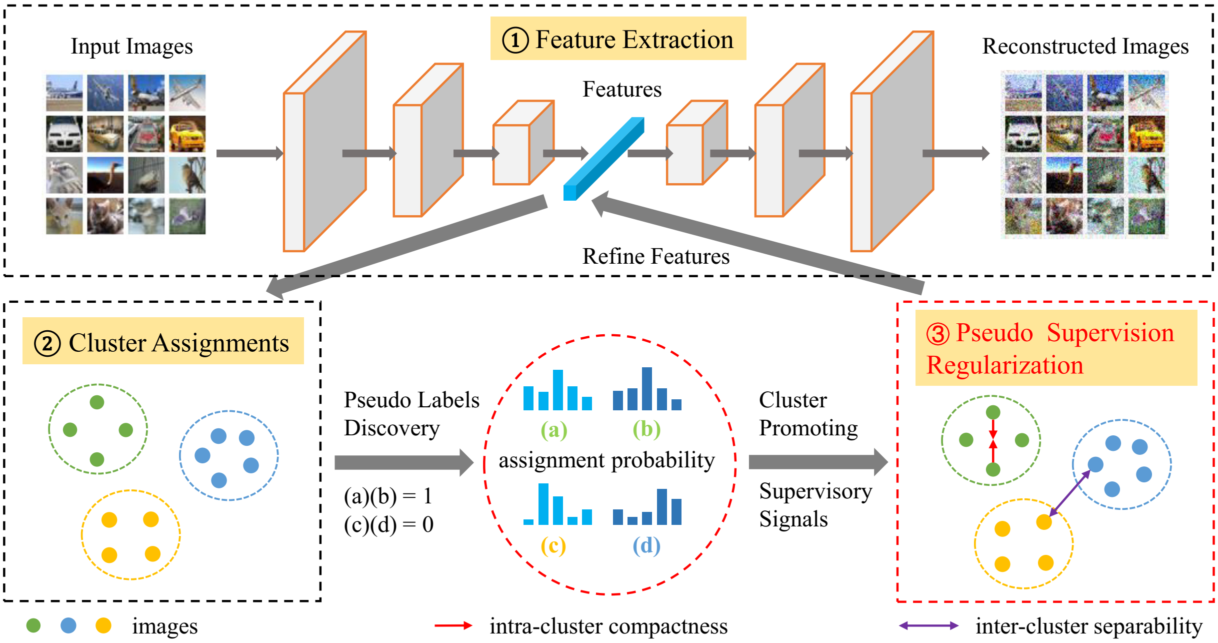


Fig. 1. The framework of our proposed DPSC model. DPSC consists of three components: (1) Feature Extraction, (2) Cluster Assignments, and (3) Pseudo Supervision Regularization. Components (1) and (2) are the basis of joint clustering methods. Component (3) is the core of DPSC. As the arrow points in the framework, our key idea is to discover the pseudo labels from clustering results as supervisory signals to facilitate inter-cluster separability and intra-cluster compactness, consequently learning more discriminative features and further enhancing the clustering performance.

Concretely, we adopt a convolutional autoencoder to extract the hierarchical feature representations of images. We compute the cluster assignment probability between samples and cluster centroids via Student's t-distribution. Most importantly, we propose to discover the pseudo labels from clustering results in hope of providing supervisory guidance for refining the representations, thereby reducing the intra-cluster distance and enlarging the inter-cluster distance progressively. To fully benefit from joint learning, we further introduce a self-evolution training algorithm to train the whole network with reconstruction loss, clustering loss, and discriminative loss jointly in an unsupervised manner. During the iteration, the learned feature representations and clustering results will constantly boost each other as more reliable pseudo supervision information is discovered. Finally, extensive experiments on different image datasets show that DPSC outperforms state-of-the-art clustering algorithms. Moreover, the learned feature representations also generalize well across various algorithms.

Our main contributions are summarized as follows:

- We propose a pseudo-supervised joint method for image clustering, namely Discriminative Pseudo Supervision Clustering (DPSC). Our key idea is to discover the pseudo supervision information to provide supervisory guidance for discriminative representation learning, hence enhancing the clustering performance.
- We further introduce a self-evolution training algorithm to jointly optimize our model, in which the learned feature representations and clustering results boost each other progressively as more reliable pseudo supervision information is discovered during the iteration.
- Extensive experiments on various image datasets show that DPSC outperforms state-of-the-art image clustering methods. Furthermore, the learned feature representations are remarkably discriminative and generalize well across various algorithms.

The remainder of this paper is organized as follows. In Section 2, we review the related work on clustering. Section 3 details our proposed model. Extensive experiments and results analysis are presented in Section 4 followed by the conclusion and future work in Section 5.

2. Related work

2.1. Clustering

Clustering algorithms can be roughly divided into four categories: hierarchial, partitional, density-based, and grid-based methods [40]. Hierarchical methods aim to merge or split data objects at each layer until the stop criteria are met. Proceeding from this definition, some essential methods are proposed, e.g., agglomerative clustering [5], hierarchical clustering [49]. As for partitional methods, the most well-known is K-means [31], which minimizes the sum of square errors between samples and their nearest cluster centroids. Density-based methods attempt to find the dense regions in a given dataset and represent them as clusters. Grid-based methods partition the data space into many cells to form a grid structure for clustering. Beyond that, there are also some other popular clustering approaches, such as spectral clustering [36], nonnegative matrix factorization based clustering [4], subspace clustering [11]. Although these traditional methods have achieved competitive success in clustering, they are primarily developed for monotonic and regular application scenarios, thus they may perform poorly in complex situations, such as high-dimensional image and text clustering. In essence, these methods perform clustering based on the original representations, thereby inheriting the limitations from the quality of data.

2.2. Feature representation learning

For image clustering, the performance highly relies on the discriminative power of extracted features. Early methods often use hand-crafted features to encode images, e.g., SIFT [29] and LBP [19]. Afterward, with the success of deep learning, various methods leveraging convolutional neural networks (CNNs) to extract deep features of images often provide state-of-the-art performance [24,26]. Nevertheless, a large amount of labeled data are required to train the network, while it is laborious and expensive to collect sufficient labeled data. To overcome this limitation, deep unsupervised representation learning, which aims to train a neural network in an unsupervised manner, has drawn much attention in recent years due to its remarkable performance [3]. Specifically, autoencoder (AE) [2] is one of the most popular methods for unsupervised feature extraction. An AE typically consists of an encoder and a decoder. It aims to minimize the reconstruction loss between inputs and outputs. Particularly, a convolutional version of AE is applied to extract hierarchical features of images [32]. Beyond that, there are also many variants of AE, such as stacked denoising autoencoder (SDAE) [39], sparse autoencoder (SAE) [33], and variational autoencoder (VAE) [23]. They are specially designed for various needs of representation learning.

2.3. Deep joint clustering

The first work to develop the autoencoder for clustering was proposed in [37], which demonstrates that the features learned by autoencoder are more suitable for clustering than raw image densities. Subsequently, DEN [21] learns the reduced representations with locality-preserving constraint and group sparsity constraint, and followed by K-means clustering. Besides, GraphEncoder [38] shares a similar idea and transfers it to graph clustering, which learns the nonlinear embeddings of the normalized graph similarity matrix by sparse autoencoder, and also adopts K-means as post-processing. Although these methods have achieved some success, there is a notable limitation, i.e., the representation learning stage and clustering stage are completely separated. Since no cluster-promoting objective is explicitly incorporated into the embedding process, the learned features may not be reliable for clustering.

To unify the processes of representation learning and clustering, DEC [43] is the pioneering method that jointly learns feature representations and cluster assignments. The joint learning process is similar to multi-objective optimization [18,17]. Concretely, DEC firstly initializes the network by pre-training the stacked denoising autoencoder. Then, DEC fine-tunes the encoder and updates the cluster assignments by minimizing the KL divergence between the soft assignments and target distribution. IDEC [15] improves DEC by preserving the reconstruction loss during clustering, leading to better performance. Similar to DEC, DCN [44] and DKM [12] jointly optimize the representation learning and K-means clustering objective to recover the clustering-friendly features. However, these methods using the fully-connected autoencoder are not suitable for handling high-dimensional structured image data. To this end, some methods [28,16,13,45] specifically leverage convolutional autoencoder (CAE) to extract hierarchical features from images and combine them with different clustering strategies, thereby improving the performance of image clustering. Among these methods, DBC [28] is exactly the convolutional version of DEC. DCEC [16] improves DEC by redesigning a strided CAE without pooling to preserve the local structure of images and keeping the decoder remained during joint training. DEPICT [13] exploits a denoising CAE and introduces the uniform prior to balance the cluster assignments by adding a regularization term into clustering loss. DDASC [45] develops a dual CAE by additionally introducing a noisy version to learn more robust representations, and inputs them to a deep spectral clustering model.

2.4. Pseudo labels

As we know, the performance of neural networks directly relies on the number of labeled examples. However, It is often quite time-consuming and expensive to collect labels in the real world. Hence, how to train neural networks with limited labels becomes an open problem. [27] proposes to train the neural network in a semi-supervised fashion by introducing pseudo labels for unlabeled data, where the pseudo labels are obtained by picking up the results with high prediction probability. This method demonstrates the effectiveness of pseudo labels. After that, various strategies have been proposed to assign the pseudo labels for unlabeled data when there are limited ground truth labels, thus enriching the training labels and enhancing the semi-supervised classification accuracy [41].

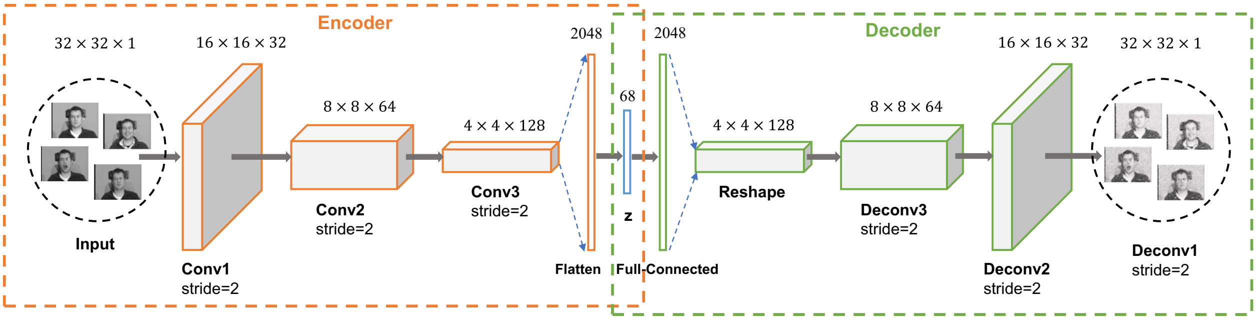
However, there are few methods to explore pseudo labels for unsupervised learning. DAC [7] recasts the clustering task as a binary pairwise-classification problem, which discovers the pairwise similarity among different samples based on the distance of label features as supervision. As an extension to DAC, DCCM [42] further utilizes the local robustness to geometry transformation and triplet mutual information between deep and shallow layers to learn better representation. However, these methods may suffer from a severe problem of error-propagation from the unreliable estimations during training, since they only use the pseudo labels related losses to train the model. For our method, the pseudo supervision is utilized as a clustering-promoting constraint enforced on the deep joint clustering framework to make the representations more discriminative.

3. Discriminative pseudo supervision clustering

Considering the clustering task of N samples $X = \{x_i\}_{i=1}^N$, the goal is to group these samples into K clusters, where each sample is denoted as $x_i \in \mathbb{R}^d$ and each cluster centroid is denoted as μ_k $(k = 1, \ldots, K)$. Instead of performing clustering directly in the data space X, we transform the raw data X into the embedded feature space $Z = \{z_i\}_{i=1}^N$, where each sample is featured as $z_i \in \mathbb{R}^e$ $(e \ll d)$.

3.1. Basic component one: image feature extraction

The fully-connected autoencoder ignores the 2D image structure information and introduces a myriad of redundant parameters, so it is not suitable for dealing with image data. To extract hierarchical features of images, we directly adopt the convolutional autoencoder (CAE) proposed in [16]. As shown in Fig. 2, the architecture of CAE is intuitively similar to autoencoder, except that the network is CNN. Concretely, CAE consists of an encoder and a decoder with a symmetric structure. The encoder tries to extract hierarchical features of the input images via several stacked convolutional layers and a flatten layer. The flattened vector is transformed into a low-dimensional representation by a fully-connected layer. On the contrary, the decoder tries to transform the embedded representation back into the original image via a fully-connected layer, a reshape layer, and several deconvolutional layers. Note that the strided convolutional layers and strided deconvolutional layers are employed to avoid deterministic spatial pooling layers (e.g., max-pooling). As a result, a better transformation capability can be provided since they allow the network to learn its own spatial downsampling [34]. Finally, CAE is trained in an unsupervised manner by minimizing the reconstruction loss:



$$L_r = \frac{1}{N}\sum_{i=1}^{N}\|x_i - g_\phi(f_\theta(x_i))\|_2^2, \tag{1}$$

where $f_\theta : X \rightarrow Z$ and $g_\phi : Z \rightarrow X$ are nonlinear mappings. θ and ϕ denote the network parameters of the encoder and decoder, respectively. Since the embedded features usually have much smaller dimensionality than the original images, it can help find the most salient features of images.

3.2. Basic component two: soft cluster aassignments

For the clustering objective, we directly follow DEC [43] to adopt the Student's t-distribution to produce the soft cluster assignments. The similarity between embedded feature z_i and centroid μ_k is then formulated as

$$p_{ik} = \frac{\left(1 + \|z_i - \mu_k\|^2\right)^{-1}}{\sum_{k'}\left(1 + \|z_i - \mu_{k'}\|^2\right)^{-1}}, \tag{2}$$

where $p_{ik} \in (0, 1)$ can be viewed as the probability of the i-th sample belonging to the k-th cluster. All p_{ik}s constitute the cluster assignment matrix $P \in \mathbb{R}^{N\times K}$. The predicted labels can be obtained from the column index of the largest entry in each row of P. In order to further improve clustering purity, [43] puts more emphasis on data points assigned with high confidence and designs a target distribution Q to refine the cluster assignment distribution P as

$$q_{ik} = \frac{p_{ik}^2/\sum_{i'}p_{i'k}}{\sum_{k'}\left(p_{ik'}^2/\sum_{i'}p_{i'k'}\right)}. \tag{3}$$

It is worth noting that q_{ik} is inclined to have stricter probability (closer to 0 and 1) by squaring the cluster assignment probability and normalizing it with the frequency of each cluster. To reduce the divergence between cluster assignment distribution P and target distribution Q, we employ the KL divergence to define the clustering loss:

$$L_c = \mathrm{KL}(Q||P) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}q_{ik}\log\frac{q_{ik}}{p_{ik}}. \tag{4}$$

3.3. Key component: pseudo supervision regularization

Jointly optimizing Eqs. (1) and (4) can obtain the final image feature representations and clustering results, which prevails in deep joint methods. However, when dealing with real-world image data, such as faces and real objects, they commonly cannot learn eminently discriminative representations, leading to entangled clusters, since they lack explicit cluster-promoting constraints on the latent representations during training. Moreover, as aforementioned, existing deep joint methods are in lack of supervision information, limiting their performance severely.

To overcome these limitations, we propose to discover the reliable pseudo labels from the cluster assignment matrix P, which is beneficial to learn more discriminative representations and thus further enhance the clustering performance. The essential insight is that benign representations are beneficial for clustering and inversely clustering results can also provide supervisory guidance for representation learning. Therefore, we exploit the pseudo labels to impose a cluster-promoting constraint on the representations explicitly. The proposed pseudo supervision regularization term is named as discriminative loss:

$$L_d = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{p}_i \cdot \hat{p}_j \cdot \left(c_{ij}\cdot r_{ij}\cdot\|z_i - z_j\|_2^2 + \left(1 - c_{ij}\cdot r_{ij}\right)\cdot \max\left(0, s - \|z_i - z_j\|_2^2\right)\right), \tag{5}$$

where c_{ij} and r_{ij} are discovered pseudo supervision information, and s is a distance constraint constant to force the distance of different clusters larger than it. $\hat{p}_i$ and $\hat{p}_j$ are the predicted cluster probabilities, i.e., the largest entry in i-th row and j-th row of P, forcing the regularization to focus on the data with high confidence. It is noted that L_d aims to reduce the intra-distance and enlarge the inter-distance, i.e., encouraging inter-cluster separability and intra-cluster compactness, which contributes to learn more discriminative representations and yields clearly separated clusters. Moreover, the regularization effect of L_d will become more and more conspicuous with the progress of clustering quality.

To discover the pseudo labels, we propose a reliable framework as shown in Fig. 3. The pseudo labels consist of two parts. One is the predicted labels c_{ij} which directly exploits the predicted labels to identify whether two images belong to the same cluster or not. Therefore, $c_{ij} = 1$ if they belong to the same cluster and $c_{ij} = 0$ otherwise. The other is pairwise patterns r_{ij} which regards the cluster assignment probability as *indicator features* of samples and aims to discover the pairwise patterns by estimating the similarities. Such a strategy is feasible based on the observation that images belonging to the same cluster will be transformed into similar representations under the same encoder, and therefore produce similar cluster assignments. As a result, the pairwise patterns can be detected as follows:

$$r_{ij} = \begin{cases} 1, & \text{if} \qquad S(p_i, p_j) > \lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where p_i and p_j correspond to the i-th row and the j-th row of P, respectively. $S(\cdot,\cdot)$ is a function to measure the similarity. Typically, we use the cosine similarity function. $\lambda \in [0,1]$ is the threshold for determining similar or dissimilar *indicator features* with high likelihood. After obtaining c_{ij} and r_{ij}, we take the product $c_{ij} \cdot r_{ij}$ as the final pseudo labels, which provides supervisory signals for the process of the entire joint learning, and forces the optimization to move in the right direction.

3.4. The objective function and optimization

Based on the above three components, the final objective function of DPSC is to minimize

$$L = L_r + L_c + \alpha L_d, \tag{7}$$

where $\alpha \in [0,1]$ controls the strength of pseudo supervision regularization. The reconstruction loss L_r aims to learn informative feature representations and prevent corrupted feature representations. The clustering loss L_c computes the cluster assignment probability and encourages clustering-friendly representations. Since both of them are the basis of deep joint clustering, hence, their coefficients are fixed at 1. More importantly, the discriminative loss L_d attempts to discover pseudo supervision information to refine the representations with inter-cluster separability and intra-cluster compactness, hence further enhancing the clustering performance.

The objective function can be optimized using stochastic gradient descent (SGD) algorithm. There are mainly two variables to update: feature representation of each image z_i and each cluster centroid μ_k. The gradients of L with respect to them are deduced as follows:

$$\begin{aligned} \frac{\partial L}{\partial z_i} &= \frac{2}{N}\sum_{k=1}^{K}(q_{ij} - p_{ij})\frac{z_i-\mu_k}{1+\|z_i-\mu_k\|^2} + \frac{1}{N^2}\sum_{j=1}^{N}\hat{p}_i \cdot \hat{p}_j \cdot \left(c_{ij}\cdot r_{ij}\cdot 2(z_i - z_j) + (1 - c_{ij}\cdot r_{ij})\cdot \max\left(0, s - 2(z_i - z_j)\right)\right), \\ \frac{\partial L}{\partial \mu_k} &= -\frac{2}{N}\sum_{i=1}^{N}(q_{ij} - p_{ij})\frac{z_i-\mu_k}{1+\|z_i-\mu_k\|^2}. \end{aligned} \tag{8}$$

Note that the gradients will be straightforwardly propagated to the network parameters with the help of backpropagation.

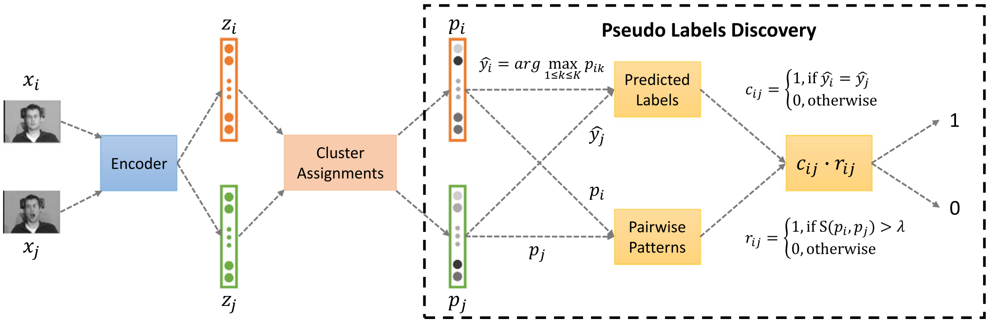

x_i
x_j
Encoder
z_i
z_j
Cluster Assignments
p_i
p_j
Pseudo Labels Discovery
$\hat{y}_i = arg \max_{1\le k\le K} p_{ik}$
Predicted Labels
$\hat{y}_j$
p_i
p_j
Pairwise Patterns
$c_{ij} = \begin{cases} 1, \text{if } \hat{y}_i = \hat{y}_j \\ 0, \text{otherwise} \end{cases}$
$r_{ij} = \begin{cases} 1, \text{if } S(p_i, p_j) > \lambda \\ 0, \text{otherwise} \end{cases}$
$c_{ij} \cdot r_{ij}$
1
0

Fig. 3. The flowchart of pseudo labels discovery. z_i and z_j are embedded features. p_i and p_j correspond to the i-th row and the j-th row of cluster assignment matrix P. The proposed pseudo labels $c_{ij} \cdot r_{ij}$ is based on the intuition that images belonging to the same clusters in ground truth should produce the same predicted labels and similar cluster assignment vectors.

3.5. Self-evolution training

To jointly learn the feature representations and cluster assignments, we develop a self-evolution training algorithm, where the ‘self-evolution’ indicates that the learned representations (more discriminative) and clustering results (more accurate) elevate each other progressively as more reliable pseudo labels are discovered during the iteration. The whole model with three components forms a closed loop and self-drives. Concretely, we firstly pre-train CAE by using reconstruction loss to obtain elementary feature representations and initialize the network parameters θ and ϕ. Then, we initialize the cluster centroid $\{\mu_k\}_{k=1}^{K}$ by performing K-means on embedded features. After initialization, we jointly train the whole network by batch with the reconstruction loss, clustering loss, and discriminative loss in an unsupervised manner. Note that the target distribution Q plays the role of ground truth but also relies on the predicted results P, so it should not be updated at each iteration using only a batch of data to avert instability. In practice, we commonly update Q after training for M iterations. The training process terminates if the change of cluster assignments between two consecutive updates for P is less than a threshold δ or the maximum number of iterations T is reached. Finally, the cluster labels can be obtained from the column index of the largest entry in each row of P. We summarize the overall training process in Algorithm 1.

```
Algorithm 1: Discriminative Pseudo Supervision Clustering
Input: Dataset $X = \{x_i\}_{i=1}^{N}$, similarity threshold $\lambda$, trade-off parameter $\alpha$, maximum iteration $T$, batch size $B$, update
   interval $M$, convergence threshold $\delta$.
Output: Cluster labels $\{l_i\}_{i=1}^{N}$.
// Pre-training stage
Pre-train CAE by Eq. (1) to initialize the network parameters $\theta$ and $\phi$.
Initialize the cluster centroid $\{\mu_k\}_{k=1}^{K}$ by performing K-means on the embedded features of pre-trained CAE.
// Joint training stage
for $iter = 1$ to $T$ do
    if $iter \% M == 0$ then
        Update the target distribution $Q$ by Eq. (3).
    // Train on batch: choose a batch of $B$ samples.
    Forward propagate to obtain embedded features $Z$.
    Compute cluster assignment matrix $P$ by Eq. (2).
    Discover pseudo labels $c_{ij}$ and $r_{ij}$ by Eq. (6).
    Compute the DPSC loss in Eq. (7).
    Backward propagate from Eq. (8) to get the gradients and update $\mu$, $\theta$, and $\phi$.
    Stop if the cluster labels $\{l_i = \arg\max_j P_{ij}\}_{i=1}^{N}$ change less than $\delta$.
end for
```

3.6. Time complexity analysis

We analyze the training and running time complexity of the proposed DPSC algorithm. The main computation includes training CAE, cluster assignments, and pseudo supervision regularization. Suppose the layer of the encoder is l, in the ith convolutional layer, the feature map size is $m_i \times m_i$, and the number of feature maps is c_i, and the kernel size of filters is $k_i \times k_i$. With the notations and assumptions, the time complexity of ith convolutional layer is $O\left(m_i^2 k_i^2 c_i c_{i+1}\right)$. Adding the time complexity of fully-connected layer, the total time complexity of training CAE is $O\left(TB\left(2\sum_i^{l-1} m_i^2 k_i^2 c_i c_{i+1} + 2c_l k_l^2 e\right)\right)$, where T is the total training iterations and B is the mini-batch size. For cluster assignments, we need to compute the cluster assignment matrix P and KL divergence loss. Its time complexity is $O(TB(Ke + e))$, where K is the number of clusters and e is the embedded dimensionality. For pseudo supervision regularization, we need to discover the pseudo labels and compute the discriminative loss. Its time complexity is $O\left(TB^2(K + e)\right)$. In real applications, e is commonly set to equal to K, and B is usually bigger than K. By summing the three items and preserving the dominant items, the overall time complexity of DPSC is approximately $O\left(TB\left(2\sum_i^{l-1} m_i^2 k_i^2 c_i c_{i+1} + 2c_l k_l^2 e + KB + eB\right)\right)$, which is comparable to existing deep joint clustering methods and is directly affected by the batch size B.

4. Experiments

In this section, we firstly evaluate the performance of the proposed model in comparison with state-of-the-art clustering methods on six popular image datasets. Then, we verify the effectiveness of the proposed pseudo supervision regularization component in improving the clustering performance and learning discriminative representations. Finally, numerous ablation study experiments are also conducted to systematically and comprehensively analyze the developed DPSC model.

4.1. Datasets

In order to evaluate the DPSC model comprehensively, we conduct experiments on various image datasets, including hand-written digit images MNIST and USPS, face images CMU-PIE and FRGC-v2.0, and real object images CIFAR-10 [13,7].

- **MNIST-full.** A 0–9 hand-written digit dataset that consists of 60000 training images and 10000 testing images with 28 by 28 pixel.
- **MNIST-test.** The testing part of MNIST-full. We regard it as a separate dataset because the number of examples may affect the clustering performance by following [16,13].
- **USPS.** A 0–9 hand-written digit dataset from the US postal service, containing 9298 images with 16 by 16 pixels.
- **CMU-PIE.** A face image dataset, including 32 by 32 face images of 68 people with 4 different expressions.
- **FRGC-v2.0.** A face image dataset. Following [13], the faces are firstly cropped and resized into 32 by 32 pixels. Then, 2462 faces are collected with 20 randomly selected subjects.
- **CIFAR-10.** A widely used real object image dataset in computer vision, containing 32 by 32 color images in 10 different classes.

Some image samples are shown in Fig. 4. The number of samples, classes, and image size are summarized in Table 1.

4.2. Evaluation metrics

To evaluate the clustering quality, we employ two widely used clustering evaluation metrics: clustering Accuracy (ACC) and Normalized Mutual Information (NMI).

4.2.1. ACC

Given a data point x_i, let r_i and g_i be the predicted label and the ground truth, respectively. The clustering accuracy is defined as

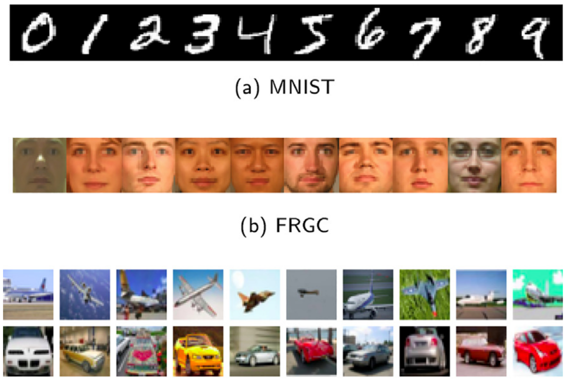
$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta(g_i, \mathrm{map}(r_i))}{n}, \tag{9}$$

where $\delta(x, y) = 1$ if $x = y$, $\delta(x, y) = 0$ otherwise, and $\mathrm{map}(r_i)$ is the permutation mapping function that maps each predicted label r_i to the equivalent label from ground truth. n is the number of samples. The optimal mapping function can be efficiently computed by Hungarian algorithm [25].

4.2.2. NMI

Let Y and C denote the set of clusters obtained from a algorithm and the ground truth, respectively. The normalized mutual information of these two sets is defined as

$$\mathrm{NMI}\ (Y, C) = \frac{\mathscr{I}(Y, C)}{[\mathscr{H}(Y) + \mathscr{H}(C)]/2}, \tag{10}$$


(a) MNIST

(b) FRGC

(c) CIFAR-10

Fig. 4. The image samples from the benchmark datasets used in our experiments.

Table 1
Statistics of datasets.

Dataset	#Domain	#Samples	#Classes	#Image size
MNIST-full	Hand-written digit	70,000	10	28 × 28 × 1
MNIST-test	Hand-written digit	10,000	10	28 × 28 × 1
USPS	Hand-written digit	9298	10	16 × 16 × 1
CMU-PIE	Face image	2856	68	32 × 32 × 1
FRGC-v2.0	Face image	2462	20	32 × 32 × 3
CIFAR-10	Real object image	60,000	10	32 × 32 × 3

where $\mathscr{I}(Y, C)$ denotes the mutual information of Y and C. $\mathscr{H}$ is the entropy of a distribution. Let n_i and $\hat{n}_j$ be the number of data in the cluster Y_i $(1 \leqslant i \leqslant c)$ and C_i $(1 \leqslant i \leqslant c)$, respectively. Denote n_{ij} as the number of data which are in the intersection between cluster Y_i and C_j. Then NMI is calculated as

$$\mathrm{NMI} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{ij} \log \frac{n_{ij}}{n_i \hat{n}_j}}{\sqrt{\left(\sum_{i=1}^{c} n_i \log \frac{n_i}{n}\right)\left(\sum_{j=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n}\right)}}. \tag{11}$$

It is easy to find that both ACC and NMI range from 0 to 1 with a higher value indicating better performance.

4.3. Comparison methods

We compare our approach with 16 clustering methods, including six baselines, eight state-of-the-art deep joint clustering methods, and two state-of-the-art pseudo labels related clustering methods. The baselines include K-means [31], spectral clustering (SC) [36], agglomerative clustering (AC) [14], large-scale spectral clustering (LSC) [9], agglomerative clustering via path integral (AC-PIC) [47], and orthogonal nonnegative graph reconstruction (ONGR) [20]. The deep joint clustering methods include deep embedded clustering (DEC) [43], improved deep embedded clustering (IDEC) [15], deep clustering network (DCN) [44], deep K-means (DKM) [12], deep spectral clustering network (SpectralNet) [35], deep discriminatively boosted clustering (DBC) [28], deep convolutional embedded clustering (DCEC) [16], and deep embedded regularized clustering (DEPICT) [13]. Among them, DEC, IDEC, DCN, DKM, and SpectralNet use a fully-connected autoencoder or neural network to learn the feature representations while DBC, DCEC, and DEPICT use a convolutional autoencoder. The pseudo labels related clustering methods include deep adaptive clustering (DAC) [7] and deep comprehensive correlation mining (DCCM) [42].

To verify the effectiveness of joint learning, we also implement a sequential method called CAE + K-means (CAEKM), which firstly employs the proposed CAE to learn the image features, and then adopts the K-means algorithm on the features to obtain the clustering results.

4.4. Experimental settings

For all datasets, we use three convolutional layers followed by a fully-connected layer in the encoder, and the decoder is a mirror of the encoder. For these three convolutional layers, the filter number is 32, 64, 128, and the kernel size is $5 \times 5, 5 \times 5, 3 \times 3$ with stride = 2. We also set proper padding to adapt images with different sizes. The dimensionality of the embedded features e and cluster number K is set to the ground truth class number of each dataset. The similarity threshold λ is tuned in [0, 1] and set to 0.9 to guarantee the pairwise patterns detected by Eq. (6) with high likelihood. The constraint constant s is set to 2 since the features have been normalized in Eq. (5). For the trade-off parameter α, we tune it in $[10^{-2}, 10^{2}]$ and set it to 0.1. To train the model, we set the batch size B to 256 and adopt Adam as our optimizer. The pre-training epochs of CAE are no more than 500. The maximum iteration T is set to 10,000 and convergence threshold δ is set to 0.001. The update interval M depends on the data size N and batch size B, which is commonly set to a value near N/B.

For all compared methods, we report the results by re-running the released code with suggested hyper-parameters. When the code is not publicly available or running the released code on large-scale datasets is not practical, we excerpt the results from their paper or put dash marks (–) instead of the corresponding results. All the experiments are repeated five times and the average results are reported.

4.5. Clustering

4.5.1. Quantitative results

Table 2 shows the quantitative results of these clustering methods on the experimental datasets. The best results are highlighted in bold fonts. For each metric, DPSC dramatically outperforms the compared algorithms on five datasets and achieves competitive results on the remaining one. Moreover, DPSC is particularly prominent on intricate images such as CMU-PIE, FRGC, and CIFAR-10. Specifically, DPSC achieves a performance promotion of 13.1% and 6.9% with respect to

ACC and NMI over the second-best results on CMU-PIE. The performance of SpectralNet and DEPICT surpass our method on USPS. However, there are conspicuous margins on other datasets. Further analysis, several conclusions can be drawn from Table 2.

- Compared with other deep joint clustering methods, the superiority of DPSC verifies that by discovering the pseudo supervision information to refine the representations and facilitate inter-cluster separability and intra-cluster compactness, DPSC can further promote the performance of joint clustering, especially for intricate images.
- Compared with other pseudo labels related clustering methods, DPSC consistently outperforms them on all datasets, which indicates the proposed pseudo supervision regularization is more effective to learn discriminative representations and is more suitable to be used as a clustering-promoting constraint enforced on the deep joint clustering framework.
- Compared with CAEKM (CAE + K-means), DPSC is significantly better than it, gaining at least 5% increment on ACC and NMI, which demonstrates that the joint methods indeed can observably improve the clustering performance, since more informative representations will be captured during clustering.

4.5.2. Generalization across clustering algorithms

We now evaluate if the feature representations learned by our method can generalize well to other clustering algorithms. We re-run all the traditional clustering methods using our learned features as inputs and keep all parameters unchanged. The results and improvements in term of ACC and NMI are reported in Table 3. It is obvious that the performance of all algorithms on most datasets has been remarkably improved compared to results in Table 2. Some classical algorithms such as K-means and AC perform poorly on the raw data space, while they have gained at most 61.4% ACC increment on CMU-PIE when they conduct clustering on our learned features. Meanwhile, the variance in performance across all baselines is much lower. For example, there is almost no performance difference for K-means, SC, and AC on MNIST-full, MNIST-test, and USPS. These results fully prove that our learned features are discriminative and applicable, also generalize well across various algorithms.

4.5.3. Effect of pseudo supervision regularization

To empirically demonstrate the importance of the pseudo supervision regularization, we perform an ablation study on the DPSC model in Eq. (7). Concretely, we directly set $\alpha = 0$ to ablate the discriminative loss (called DPSC *w/o* PS, i.e., without pseudo supervision). Furthermore, we also ablate the reconstruction loss (called DPSC *w/o* RL, i.e., without the decoder of CAE) to make a more sufficient comparison. Other experimental settings inherit from Section 4.4. Table 4 reports the numerical experimental results and Fig. 5 visualizes them. From the results, there are two observations. (1) Compared with DPSC *w/o* PS and DPSC, the proposed pseudo supervision regularization significantly helps to improve the clustering performance, and the improvements are more obvious on the intricate images, such as face images CMU-PIE, FRGC, and real object images CIFAR-10. (2) Compared with DPSC *w/o* PS and DPSC *w/o* RL, the discriminative loss prominently contributes more to improve the clustering performance than the reconstruction loss, since it acts as a clustering-promoting constraint that directly helps to learn more discriminative representations.

Table 2
Clustering results of various methods on six image datasets. Results marked with † are excerpted from their paper. Dash marks (–) indicates the results are unavailable.

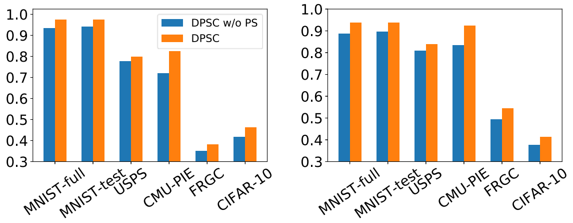
Dataset	MNIST-full		MNIST-test		USPS		CMU-PIE		FRGC		CIFAR-10	
Metric	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
K-means	0.535	0.500	0.554	0.513	0.672	0.614	0.208	0.421	0.244	0.280	0.229	0.087
SC	0.663	0.764	0.650	0.715	0.677	0.814	0.496	0.680	0.270	0.359	0.247	0.103
AC	0.621	0.682	0.695	0.711	0.653	0.722	0.258	0.524	0.256	0.330	0.228	0.105
AC-PIC	–	–	0.850	0.844	0.660	0.804	0.480	0.692	0.304	0.415	–	–
LSC	0.576	0.628	0.390	0.324	0.610	0.695	0.432	0.651	0.308	0.416	0.227	0.101
ONGR	0.606	0.637	0.454	0.435	0.642	0.701	0.467	0.692	0.333	0.428	0.238	0.115
DEC	0.830	0.832	0.839	0.848	0.773	0.794	0.324	0.676	0.270	0.326	0.301	0.257
IDEC	0.880	0.869	0.878	0.868	0.788	0.769	0.354	0.655	0.328	0.422	0.325	0.276
DCN	0.840	0.800	0.837	0.803	0.660	0.657	0.102	0.298	0.161	0.150	0.210	0.163
DKM	0.840†	0.796†	–	–	0.757†	0.776†	–	–	–	–	–	–
SpectralNet	0.969	0.922	0.956	0.898	0.817	**0.862**	0.504	0.762	0.288	0.340	0.327	0.279
DBC	0.874	0.858	0.877	0.864	0.659	0.648	0.486	0.713	0.286	0.382	0.315	0.265
DCEC	0.896	0.890	0.858	0.841	0.798	0.845	0.693	0.856	0.353	0.476	0.405	0.347
DEPICT	0.965	0.917	0.963	0.915	**0.838**	**0.862**	0.219	0.457	0.264	0.279	0.351	0.294
DAC	0.974	0.934	0.960	0.928	0.614	0.628	0.172	0.405	0.185	0.173	0.225	0.113
DCCM	0.955	0.919	0.937	0.906	0.686	0.675	0.235	0.522	0.289	0.312	0.371	0.294
CAEKM	0.921	0.841	0.931	0.862	0.731	0.699	0.652	0.790	0.317	0.415	0.376	0.311
DPSC	**0.976**	**0.939**	**0.976**	**0.939**	0.800	0.840	**0.824**	**0.925**	**0.381**	**0.544**	**0.464**	**0.413**

Table 3
Clustering results and improvements (compared to Table 2) of traditional methods using our learned features as inputs.

Dataset	MNIST-full		MNIST-test		USPS		CMU-PIE		FRGC	
Metric	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
K-means	0.976	0.939	0.976	0.939	0.800	0.840	0.822	0.925	0.340	0.498
Improvements (%)	(+44.1)	(+43.9)	(+42.2)	(+42.6)	(+12.8)	(+22.6)	(+61.4)	(+50.4)	(+9.6)	(+21.8)
SC	0.976	0.939	0.976	0.940	0.800	0.838	0.377	0.712	0.258	0.344
Improvements (%)	(+31.3)	(+17.5)	(+32.6)	(+22.5)	(+12.3)	(+2.4)	(−11.9)	(+3.2)	(−1.2)	(−1.5)
AC	0.975	0.937	0.973	0.935	0.799	0.838	0.833	0.927	0.340	0.497
Improvements (%)	(+35.4)	(+25.5)	(+27.8)	(+22.4)	(+14.6)	(+11.6)	(+57.5)	(+40.3)	(+8.4)	(+16.7)
AC-PIC	-	-	0.601	0.771	0.874	0.861	0.814	0.915	0.346	0.498
Improvements (%)	-	-	(−24.9)	(−7.3)	(+21.4)	(+5.7)	(+33.4)	(+22.3)	(+4.2)	(+8.3)
LSC	0.657	0.754	0.708	0.777	0.605	0.686	0.778	0.906	0.327	0.441
Improvements (%)	(+8.1)	(+12.6)	(+31.8)	(+45.3)	(−0.5)	(−0.9)	(+34.6)	(+25.5)	(+1.9)	(+2.5)
ONGR	0.808	0.882	0.85	0.896	0.746	0.818	0.649	0.880	0.325	0.470
Improvements (%)	(+20.2)	(+24.5)	(+39.6)	(+46.1)	(+10.4)	(+11.7)	(+18.2)	(+18.8)	(−0.8)	(+4.2)

Table 4
Effect of pseudo supervision regularization. PS denotes pseudo supervision. RL denotes reconstruction loss.

Dataset	MNIST-full		MNIST-test		USPS		CMU-PIE		FRGC		CIFAR-10	
Metric	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
DPSC *w/o* PS	0.934	0.888	0.941	0.897	0.778	0.810	0.720	0.835	0.350	0.494	0.419	0.378
DPSC *w/o* RL	0.975	0.937	0.975	0.937	0.793	0.83	0.810	0.921	0.371	0.531	0.453	0.406
DPSC	**0.976**	**0.939**	**0.976**	**0.939**	**0.800**	**0.840**	**0.824**	**0.925**	**0.381**	**0.544**	**0.464**	**0.413**



(a) ACC (b) NMI

Fig. 5. Effect of pseudo supervision: clustering performance comparison of DPSC *w/o* PS (without pseudo supervision) and DPSC on various datasets.

4.5.4. Discriminability proof of learned features

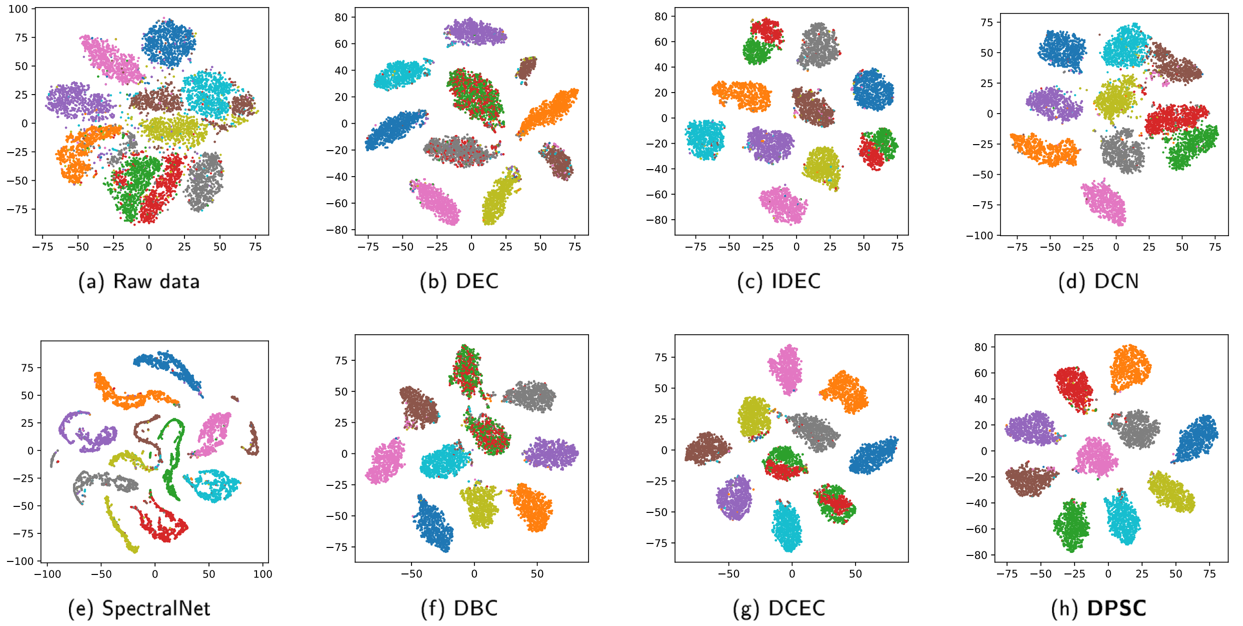
To intuitively illustrate the final features learned by various deep clustering methods, taking MNIST-full as an example, we firstly map the features into a 2-dimensional space with t-SNE [30] and then visualize a randomly sampled subset of 10 K data points in Fig. 6, where different colors correspond to different clusters. As can be seen, the raw pixel features are mixed and have no clear cluster structure. The features learned by other deep joint methods are still entangled in some similar hand-written digits, e.g., red cluster (digit 4) and green cluster (digit 9), or have smaller inter-cluster separability, which will lead to unstable clusters and thus deteriorate the clustering performance. It is worth noting that the features learned by our method are exceedingly discriminative with larger inter-cluster separability and intra-cluster compactness and possess correctly separated clusters even for similar hand-written digits. Moreover, the clusters strongly correspond to the ground truth but are discovered with no supervision.

4.5.5. Visualization of training process

In Fig. 7, we visualize the progression of the embedded features and clustering performance variation on USPS during training. It is clear that our method gradually produces more accurate cluster assignments and the clusters are becoming increasingly well separated. Fig. 7(d) shows how clustering ACC and NMI correspondingly increase and become stable over training epochs. Similar results can be observed on other datasets. It is worth noting that ACC and NMI converge to promising values quickly with less than 20 epochs.

4.5.6. Running time comparison

In Section 3.6, we analyze the time complexity of the proposed model theoretically. Now we compare the practical running time of DPSC and some representative deep joint clustering methods. We run DPSC and the released code of other methods on a machine with one NVIDIA GTX 1070Ti GPU and an Intel Core i5-7500 CPU (3.4 GHz). Fig. 9 illustrates the results. As

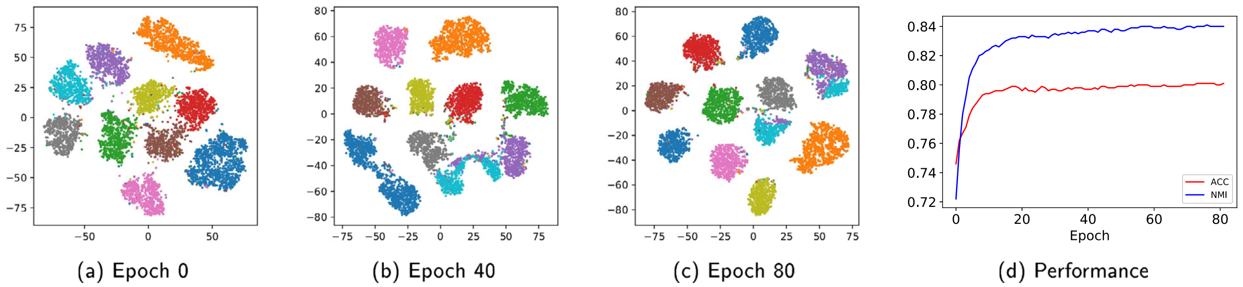


(a) Raw data (b) DEC (c) IDEC (d) DCN

(e) SpectralNet (f) DBC (g) DCEC (h) **DPSC**

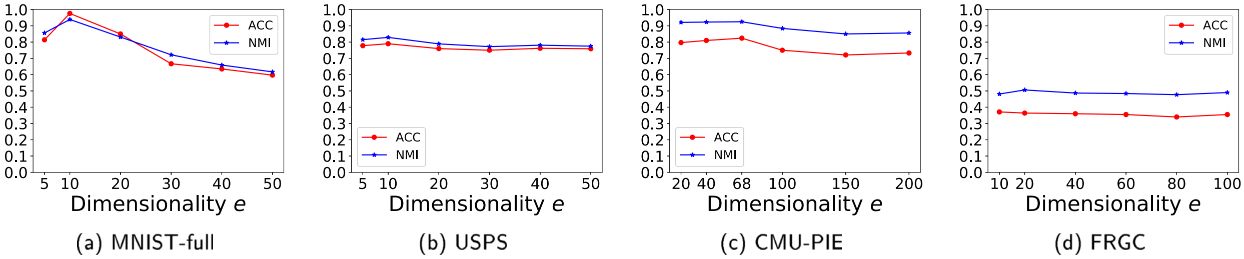
Fig. 6. Discriminability proof: t-SNE visualization of the final features learned by different methods on MNIST-full with randomly sampled 10 K data points, where different colors correspond to different clusters. The features learned by DPSC are the most discriminative. Moreover, better discriminability of the learned features should be observed on intricate images, since more prominent performance improvements happened to them.

can be seen, although the running time of DPSC is slightly higher than DBC and DCEC due to the proposed pseudo supervision regularization component, the improvement in ACC and NMI are significant in Table 2. Moreover, when dealing with the larger datasets (MNIST-full), DPSC still shows acceptable running time, but the running time of DEPICT dramatically grows with the size of input data. This outcome indicates the practicality of DPSC for real-world clustering tasks.



(a) Epoch 0 (b) Epoch 40 (c) Epoch 80 (d) Performance

Fig. 7. Training process: visualization of the embedded features (a)–(c) and clustering ACC and NMI (d) change with training epochs on USPS.



(a) MNIST-full (b) USPS (c) CMU-PIE (d) FRGC

Fig. 8. Embedded dimensionality: comparison of clustering performance with different embedded dimensionalities.

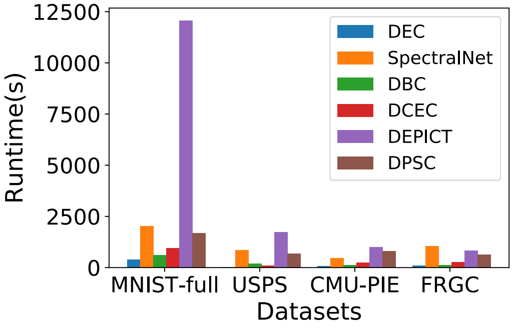


Fig. 9. Runtime comparison: Running time of various deep joint clustering methods on different image datasets.

4.6. Ablation study

4.6.1. Robustness to embedded dimensionality

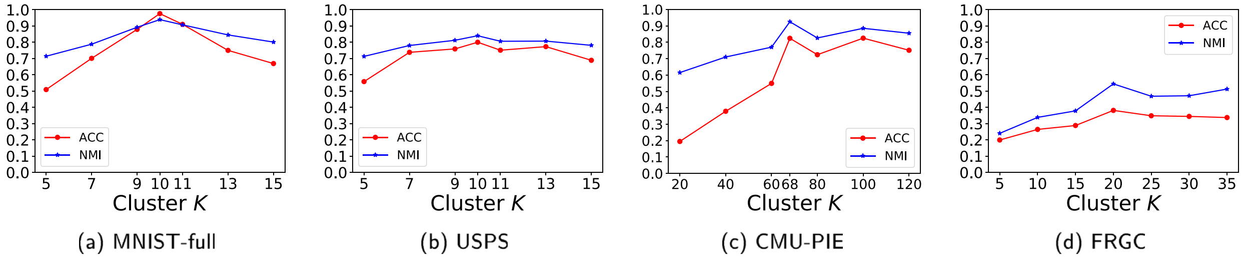
We conduct an experiment to study the robustness of DPSC by varying the dimensionalities of embedded features. We set different dimensionality ranges for datasets with different cluster numbers. The results are shown in Fig. 8,9. There are two observations. (1) The promising results are achieved when the dimensionalities are set to the cluster numbers of datasets. (2) There is almost no large performance fluctuation across these datasets except MNIST-full, which demonstrates that DPSC is robust to the dimensionalities of embedded features.

4.6.2. Impact of cluster number

For practical clustering tasks, we may not know the specific cluster number K. Therefore, we conduct an experiment to study the stabilities of DPSC by varying the number of clusters. We set different ranges of K according to the ground truth class number of different datasets. The results are shown in Fig. 10. As the number of clusters increases, the performance of DPSC first increases and then decreases gently, and the best performance is obtained when K is equal to the ground truth class number. It is noted that DPSC achieves comparable performance when K is bigger than the ground truth class number, which implies DPSC has the capability to deal with clustering tasks with a large cluster number.

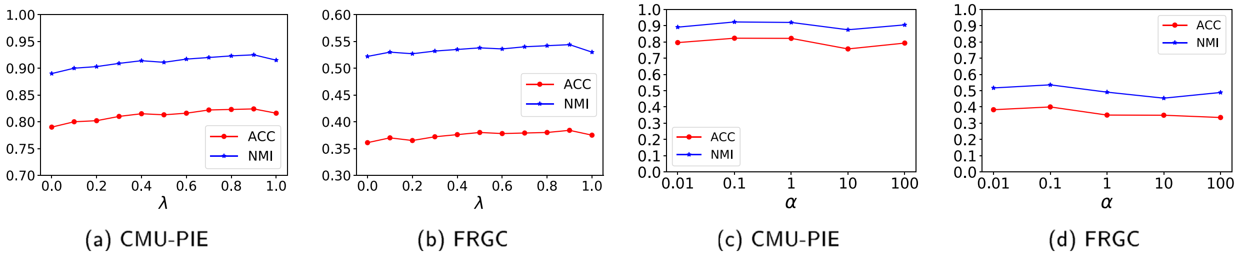
4.7. Parameter sensitivity analysis

We further study how our method performs when using different settings of parameters. There are two tuning parameters in our model, i.e., similarity threshold λ and trade-off parameter α. For parameter λ, it ranges from 0 to 1, deciding whether the pairwise pattern r_{ij} is 0 or 1. We tune λ in [0, 1] with step size 0.1. For parameter α, it controls the contributions of discriminative representation learning. We tune α in the range of $\left\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\right\}$. The effect of α and λ on CMU-PIE and FRGC is shown in Fig. 11. Similar results can be observed on other datasets. As we can see from Fig. 11(a)(b), with the increase of λ, ACC and NMI first increase and then decrease. It is consistent with our claim that λ should be slightly large to guarantee the pairwise patterns r_{ij} detected by Eq. (6) with high likelihood. Since the proposed pseudo labels consist of two parts, i.e., $c_{ij} \cdot r_{ij}$, the performance is relatively stable and controllable even when λ is small. As we can see from Fig. 11(c)(d), our method maintains acceptable results with different magnitude values of α and the promising results are achieved when α is relatively small (e.g., $\alpha = 0.1$). It demonstrates that the proposed pseudo supervision component can indeed be treated as an effective cluster-promoting regularizer for the joint clustering problem.



(a) MNIST-full (b) USPS (c) CMU-PIE (d) FRGC

Fig. 10. Cluster number: comparison of clustering performance with different number of clusters.



(a) CMU-PIE (b) FRGC (c) CMU-PIE (d) FRGC

Fig. 11. Parameter analysis: clustering results with different values of λ and α on CMU-PIE and FRGC.

5. Conclusion

In this paper, we propose a pseudo-supervised joint method DPSC for image clustering, which paves a new way for the deep clustering problem. Instead of focusing on new combination strategies of neural networks and clustering-oriented objectives, DPSC treats feature extraction and cluster assignments as the basic component of joint clustering, putting primary attention on discovering the pseudo supervision information from clustering results so as to enhance the discriminability of image representations. With the aid of exploratory pseudo supervision information, DPSC is capable of learning more discriminative representations with inter-cluster separability and intra-cluster compactness. Meanwhile, the learned representations and clustering results progressively promote each other as more reliable pseudo labels are discovered during the iteration. Finally, experiments on six popular image datasets demonstrate that DPSC achieves superior performance as compared to state-of-the-art methods. Moreover, the image feature representations learned by DPSC are remarkably discriminative and generalize well across various algorithms. Therefore, our method can also be treated as an unsupervised representation learning approach for images.

The proposed pseudo supervision regularization is not limited to the DPSC model, which is exactly a general promoted component for the joint clustering framework to learn more discriminative representations. In the future, we are interested in exploring the effectiveness of pseudo supervision on other unsupervised methods and tasks.

CRediT authorship contribution statement

Weibo Hu: Conceptualization, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Chuan Chen:** Conceptualization, Methodology, Project administration, Writing - review & editing. **Fanghua Ye:** Conceptualization, Methodology, Writing - review & editing. **Zibin Zheng:** Supervision, Funding acquisition. **Yunfei Du:** Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The research is supported by the Key-Area Research and Development Program of Guangdong Province (No. 2020B010165003), the National Natural Science Foundation of China (No. 11801595), the Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515011043), the Natural Science Foundation of Guangdong (No. 2018A030310076) and the CCF-Tencent Open Fund WeBank Special Funding.

Appendix A

A.1. Clustering results on noisy image data

In the real world, images are not completely clean and may contain noise, which directly affects the clustering results. Hence, we add an experiment to evaluate if the proposed method has the robustness to these kinds of images.

A.1.1. Synthetic noise image

Taking MNIST-test dataset as an example, we contaminate the images by using two typical kinds of additive noise, random pixel corruption, and Gaussian noise. For random pixel corruption, we directly replace randomly selected pixels with values under a uniform distribution on [0,255]. For Gaussian noise, we add the value sampled from a Gaussian distribution with a mean of 0 and a std of 255 to the randomly selected pixels. The noise ratio, i.e., the percentage of randomly selected

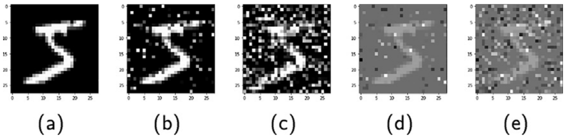


Fig. 12. Example of noisy images on MNIST-test dataset. (a) Original image, (b/c) 10%/40% noisy by random pixel corruption, (d/e) 10%/40% noisy by Gaussian noise.

pixels, is set to 10% and 40% to simulate a light and heavy contamination, respectively. Note that the ratio is lower than 50% to guarantee that the model would not confuse the noise with clean pixels. Some noisy images are shown in Fig. 12.

A.1.2. Experimental settings

To fully explore the robustness of our proposed method DPSC to noisy image data, we implement DPSC with four versions. The main difference lies in the input and ground truth of CAE (see Fig. 2) in the pre-training stage. Table 5 shows the difference. The details are depicted as follows:

- **DPSC *w/o* PS (using CAE).** The ablation version of DPSC, i.e., without pseudo supervision. It pre-trains CAE with noisy images as inputs and ground truth.
- **DPSC (using CAE).** It pre-trains CAE with noisy images as inputs and ground truth.
- **DPSC (using denoised CAE).** It replace CAE with denoised CAE to extract the features of noisy images. Denoised CAE receives the noisy image as input but uses the clean image as ground truth to compute the reconstruction loss. In this way, the network is forced to learn robust parameters so that it can absorb the noise from the inputs.
- **DPSC (using clean CAE).** It pre-trains CAE with clean images as inputs and ground truth.

In the joint training stage, all the methods take the noisy images as the input and ground truth of CAE. Other experimental settings inherit from Section 4.4.

A.1.3. Quantitative results

The clustering results on MNIST-test with different noise types and ratios are reported in Table 6. From the results, we have the following observations:

Table 5
Explanation of different DPSC versions. X denotes the clean images, and $\tilde{X}$ denotes the noisy images. Pseudo supervision, i.e., the proposed pseudo supervision regularization or discriminative loss. CAE is the abbreviation of convolutional autoencoder. The ground truth is used to compute the construction loss of CAE.

Method	Pseudo supervision	CAE in pre-training stage		CAE in joint training stage	
		Input	Ground truth	Input	Ground truth
DPSC *w/o* PS (using CAE)		$\tilde{X}$	$\tilde{X}$	$\tilde{X}$	$\tilde{X}$
DPSC (using CAE)	√	$\tilde{X}$	$\tilde{X}$	$\tilde{X}$	$\tilde{X}$
DPSC (using denoised CAE)	√	$\tilde{X}$	X	$\tilde{X}$	$\tilde{X}$
DPSC (using clean CAE)	√	X	X	$\tilde{X}$	$\tilde{X}$

Table 6
Clustering results on MNIST-test with different noise types and ratios.

Method	Random pixel corruption				Gaussian noise			
	10%		40%		10%		40%	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
DPSC *w/o* PS (using CAE)	0.846	0.822	0.796	0.749	0.781	0.754	0.66	0.653
DPSC (using CAE)	0.921	0.882	0.817	0.772	0.846	0.814	0.735	0.723
DPSC (using denoised CAE)	0.927	0.901	**0.832**	**0.804**	0.863	0.835	0.775	0.748
DPSC (using clean CAE)	**0.972**	**0.93**	0.787	0.772	**0.963**	**0.911**	**0.793**	**0.792**

- Even pre-training CAE with the noisy images as inputs and ground truth, the performance of DPSC (using CAE) is acceptable and distinctly superior to DPSC *w/o* PS (using CAE). It indicates the proposed pseudo supervision component contributes to improving the model robustness. One reliable reason is that the pseudo supervision component helps to learn more discriminative features, and thus DPSC is not easily affected by the noise in the input.
- When there is a small amount of noise in the images, such as the 10% ratio, DPSC (using clean CAE) achieves promising performance with almost no degradation. It indicates DPSC is capable of handling small noisy images when a clean initialization weight is assigned.
- When there is a lot of noise in the images, such as the 40% ratio, we can use the denoised CAE to explicitly alleviate the impact of noise, which can further improve the robustness of DPSC.

References

[1] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, D. Cremers, Clustering with deep learning: taxonomy and new methods, 2018. arXiv preprint arXiv:1801.07648.
[2] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, ICML Workshop (2012) 37–49.
[3] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.
[4] D. Cai, X. He, X. Wang, H. Bao, J. Han, Locality preserving nonnegative matrix factorization, in: Twenty-First International Joint Conference on Artificial Intelligence, 2009.
[5] Z. Cai, X. Yang, T. Huang, W. Zhu, A new similarity combining reconstruction coefficient with pairwise distance for agglomerative clustering, Inf. Sci. 508 (2020) 173–182.
[6] S. Chakraborty, S. Das, Detecting meaningful clusters from high-dimensional data: a strongly consistent sparse center-based clustering approach, in: IEEE Trans. Pattern Anal. Mach. Intell., 2020.
[7] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep adaptive image clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5879–5887.
[8] B. Chen, W. Deng, Energy confused adversarial metric learning for zero-shot image retrieval and clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 8134–8141.
[9] X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, in: AAAI, AAAI Press, 2011, pp. 313–318.
[10] J. Cheng, L. Ye, Y. Guo, J. Zhang, H. An, Ground crack recognition based on fully convolutional network with multi-scale input, IEEE Access 8 (2020) 53034–53048.
[11] T. Deng, D. Ye, R. Ma, H. Fujita, L. Xiong, Low-rank local tangent space embedding for subspace clustering, Inf. Sci. 508 (2020) 1–21.
[12] M.M. Fard, T. Thonet, E. Gaussier, Deep k-means: Jointly clustering with k-means and learning representations, 2018. arXiv preprint arXiv:1806.10069..
[13] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, ICCV (2017) 5736–5745.
[14] K.C. Gowda, G. Krishna, Agglomerative clustering using the concept of mutual nearest neighbourhood, Pattern Recogn. 10 (1978) 105–112.
[15] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: IJCAI, 2017, pp. 1753–1759..
[16] X. Guo, X. Liu, E. Zhu, J. Yin, Deep clustering with convolutional autoencoders, in: ICNIP, Springer, 2017, pp. 373–382.
[17] Y. Guo, H. Yang, M. Chen, J. Cheng, D. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm Evol. Comput. 48 (2019) 156–171.
[18] Y.N. Guo, X. Zhang, D.W. Gong, Z. Zhang, J.J. Yang, Novel interactive preference-based multi-objective evolutionary optimization for bolt supporting networks, IEEE Trans. Evol. Comput. (2019).
[19] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. Image Process. 19 (2010) 1657–1663.
[20] J. Han, K. Xiong, F. Nie, Orthogonal and nonnegative graph reconstruction for large scale clustering, in: IJCAI, 2017, pp. 1809–1815..
[21] P. Huang, Y. Huang, W. Wang, L. Wang, Deep embedding network for clustering, in: ICPR, IEEE, 2014, pp. 1532–1537.
[22] Z. Kang, X. Lu, J. Liang, K. Bai, Z. Xu, Relation-guided representation learning, Neural Networks 131 (2020) 93–102.
[23] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013. arXiv preprint arXiv:1312.6114..
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[25] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logist. (NRL) 52 (2005) 7–21.
[26] A. Kumar, S.K. Singh, S. Saxena, K. Lakshmanan, A.K. Sangaiah, H. Chauhan, S. Shrivastava, R.K. Singh, Deep feature learning for histopathological image classification of canine mammary tumors and human breast cancer, Inf. Sci. 508 (2020) 405–421.
[27] D.H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on challenges in representation learning, ICML, 2013..
[28] F. Li, H. Qiao, B. Zhang, Discriminatively boosted image clustering with fully convolutional auto-encoders, Pattern Recogn. 83 (2018) 161–173.
[29] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2004) 91–110.
[30] L.V.D. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[31] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 1967, pp. 281–297..
[32] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: ICANN, Springer, 2011, pp. 52–59..
[33] A. Ng et al, Sparse autoencoder, CS294A Lecture Notes 72 (2011) 1–19.
[34] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. arXiv preprint arXiv:1511.06434..
[35] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, Y. Kluger, Spectralnet: Spectral clustering using deep neural networks, 2018. arXiv preprint arXiv:1801.01587.
[36] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[37] C. Song, F. Liu, Y. Huang, L. Wang, T. Tan, Auto-encoder based data clustering, in: CIARP, Springer, 2013, pp. 117–124.
[38] F. Tian, B. Gao, Q. Cui, E. Chen, T.Y. Liu, Learning deep representations for graph clustering, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.
[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[40] S. Wazarkar, B.N. Keshavamurthy, A survey on image data analysis through clustering techniques for real world applications, J. Vis. Commun. Image Represent. 55 (2018) 596–626.
[41] H. Wu, S. Prasad, Semi-supervised deep learning using pseudo labels for hyperspectral image classification, IEEE Trans. Image Process. 27 (2017) 1259–1270.

[42] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, H. Zha, Deep comprehensive correlation mining for image clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8150–8159.

[43] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, ICML (2016) 478–487.

[44] B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong, Towards k-means-friendly spaces: simultaneous deep learning and clustering, in: ICML, JMLR.org, 2017, pp. 3861–3870..

[45] X. Yang, C. Deng, F. Zheng, J. Yan, W. Liu, Deep spectral clustering using dual autoencoder networ, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4066–4075.

[46] Y. Yang, D. Xu, F. Nie, S. Yan, Y. Zhuang, Image clustering using local discriminant models and global integration, IEEE Trans. Image Process. 19 (2010) 2761–2773.

[47] W. Zhang, D. Zhao, X. Wang, Agglomerative clustering via maximum incremental path integral, Pattern Recogn. 46 (2013) 3056–3065.

[48] X. Zhang, Y. Sun, H. Liu, Z. Hou, F. Zhao, C. Zhang, Improved clustering algorithms for image segmentation based on non-local information and back projection, Inf. Sci. 550 (2021) 129–144.

[49] F. Zhou, F.D. la Torre, J.K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 582–596.