Knowledge-Based Systems 253 (2022) 109507

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Improved deep clustering model based on semantic consistency for image clustering

Feng Zhang [a], Lin Li [a], Qiang Hua [a,*], Chun-Ru Dong [a], Boon-Han Lim [b]

[a] Hebei Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, China
[b] Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Kajang, Malaysia

ARTICLE INFO

Article history:
Received 22 March 2022
Received in revised form 19 July 2022
Accepted 20 July 2022
Available online 25 July 2022

Keywords:
Contrastive learning
Deep clustering
Semantic consistency
Image clustering

ABSTRACT

Recently, contrastive learning has gained increasing attention as a research topic for image-clustering tasks. However, most contrastive learning-based clustering models focus only on the similarity of embedded features or divergence of cluster assignments, without considering the semantic distribution of instances, undermining the performance of clustering. Therefore, an improved deep clustering model based on semantic consistency (DCSC) was proposed in this study, motivated by the assumption that the semantic probability distribution of various augmentations of the same instance should be similar and that of different instances should be orthogonal. The DCSC fully exploits instance-level differentiation, cluster-level discrimination, and semantic consistency of instances to design the objective function. Compared with existing contrastive learning-based clustering models, the proposed model is more cluster-sensitive to differentiate semantic concepts owing to the incorporation of cluster structure discovering loss. Extensive experimental results on six benchmark datasets illustrate that the proposed DCSC achieves superior performance compared to the state-of-the-art clustering models, with an improved accuracy of 9.3% for CIFAR-100 and 22.1% for tiny-ImageNet. The visualization results show that the DCSC produces geometrically well-separated cluster embeddings defined by the Euclidean distance, verifying the effectiveness of the proposed DCSC.

1. Introduction

Image clustering and image classification are typical processes that use machine learning models, which have boosted various applications in data mining, pattern recognition, and computer vision [1–4]. However, image clustering is more challenging than image classification because no supervised information is available. It is hard to leverage widely used traditional clustering methods, such as K-means [5], spectral clustering [6], and agglomerative clustering [7], owing to the curse of dimensionality caused by image data.

To overcome the curse of dimensionality, many researchers have adopted embedding-based methods to reduce the dimensions of input images by mapping the high-dimensional input into a low-dimensional feature embedding space. Earlier embedding-based methods employed various handcrafted features, such as SIFT [8] and HOG [9], to obtain invariant embedded features at the cost of high computational complexity. In the past decade, combining deep convolutional neural networks with traditional clustering algorithms to perform image clustering has become popular. Numerous deep-clustering methods have been proposed [6,7,10–29]. The existing deep clustering models can be classified into two categories.

(1) ***Alternating-training-based clustering method,*** a two-stage deep clustering model, has been proposed by iteratively estimating the cluster assignment and updating the model parameters. The main idea underlying this method is iteratively employing traditional clustering algorithms to group the embedded features extracted by current deep CNNs and utilizing its estimated cluster assignment to update the parameters of the deep model [6,7,10–17,29]. Although this method has achieved encouraging performance, it often suffers from error accumulation due to the separation of the learning and clustering phases, undermining the performance of clustering. In addition, it is also difficult to learn a discriminative representation that is beneficial for discovering the boundaries of the inherent cluster because entangled feature embeddings might be learned due to the intrinsic defects of autoencoders [30–32].

(2) ***Joint-training-based clustering method,*** an end-to-end deep clustering model, was proposed to alleviate the aforementioned error accumulation problem. This type of method can simultaneously learn feature representations and estimate the cluster assignment of input data [18–21,33–36]. Most clustering

[*] Corresponding author.
E-mail address: huaq@hbu.cn (Q. Hua).

https://doi.org/10.1016/j.knosys.2022.109507
0950-7051/© 2022 Elsevier B.V. All rights reserved.

models are driven by contrastive learning, which was inspired by the pioneering study of Becker and Hinton [37]. Most models neglect the semantic similarities of instances by simply pushing embedded features apart as long as they are from different instances, although these joint-learning-based clustering algorithms have achieved great success. This might lead to unstable clustering performance due to the ignorance of semantic consistency among instances, giving rise to worse clustering performance in some cases [18–21,38].

An improved deep clustering algorithm based on semantic consistency (DCSC) is proposed in this study, motivated by the assumption that the semantic cluster assignment for the same instances with different augmentations should be the same and that of different instances should be orthogonal. In contrast to existing contrastive-based clustering methods, DCSC imposes cluster structure discovery during the learning process and consequently endows contrastive learning-based clustering models with the capability of high-level cluster understanding. Therefore, more semantic information from embedded features can be learned. The main contributions of this study are summarized as follows.

(1) A deep image clustering model that accommodates representation learning and image clustering concurrently in an end-to-end model is proposed by incorporating a cluster structure that discovers loss into conventional contrastive learning-based clustering objective functions to learn semantic cluster boundaries.

(2) The DCSC addresses not only the instance-level differentiation in the feature embedding space but also the cluster-level discrimination and semantic consistency of instances in the whole batch to implicitly model the inter-cluster and intra-cluster decision boundaries.

(3) Extensive experiments demonstrate that DCSC performs better than most existing image clustering methods, with an improvement of 9.3% and 22.1% in accuracy on CIFAR-100 and tiny-ImageNet, respectively. Various ablation studies have been performed, and some valuable insights have been derived from the visualization results.

The remainder of this paper is organized as follows. In Section 2, a brief overview of the related models and discussions are presented. In Section 3, a deep clustering model based on semantic-consistent contrastive learning clustering is described in detail and its corresponding algorithm is introduced. Section 4 presents the experimental setup and the comparison results. The experimental results are thoroughly analyzed and discussed. Finally, in Section 5, the conclusions and suggestions for future studies are provided.

2. Related work

Traditional clustering approaches struggle to deal with high-dimensional image data. Therefore, various deep CNN-based image-clustering algorithms have been proposed. According to the training strategy, these deep CNN-based image clustering models can be divided into two groups: (1) alternating-training-based models [6,7,10–14,22–26] and (2) joint-training-based models [18–21,33,38].

2.1. Alternating-training-based clustering models

The alternating-training-based clustering model performs clustering in a two-stage manner. First, it extracts the feature representation using a deep CNN model, and then, it leverages the learned features to estimate the cluster assignment. It switches back and forth between training the deep model and predicting cluster assignments. For example, deep embedding clustering (DEC) [12] used stacked autoencoders to map high-dimensional images into a low-dimensional space and then leveraged traditional K-means [5] to perform clustering. The stacked autoencoders were iteratively updated based on the estimated cluster assignments. Deep embedded regularized clustering (DEPICT) [16] employed an alternating strategy to update the network parameters and estimated cluster assignments by optimizing a regularized clustering objective function. JULE [23] used an agglomerative clustering method to obtain clustering assignments based on current feature representations and used the estimated clustering assignments to update the parameters of the network alternately. DCCM [25] proposed triplet mutual information among features to comprehensively mine the relationship between the deep- and shallow-layer representations of each instance for better cluster assignment estimations.

These two-stage methods have achieved great success in image clustering by enhancing the quality of the feature representation. However, it is possible for the errors to accumulate gradually in the alternate updating of the feature learning stage and the cluster assignment stage, reducing image clustering performance [10–14,22–25], owing to the independently separated process of representation learning and clustering. In addition, these methods are only applicable to offline tasks, limiting their use in large-scale online image-clustering scenarios. Consequently, many researchers have focused on clustering models based on joint training strategies.

2.2. Joint-training-based clustering models

These models incorporate feature representation and clustering into an end-to-end framework and simultaneously update the feature representation and clustering heads. The feature representation head provides feature embeddings with fine-grained information, whereas the clustering head predicts the cluster assignments. End-to-end clustering models face the challenge of the representation collision problem because they make a prediction directly in the feature embedding space. Numerous contrastive learning-based clustering models have been proposed for image clustering [18–21,31,33,35,36,39–46] based on the observation that contrastive loss can circumvent collapsing problems in clustering by formulating the prediction problems into discrimination tasks [44]. For example, the invariant information clustering (IIC) model [18] was proposed to learn invariant features from different augmented images and performed clustering by maximizing the mutual information between class assignments of each paired instance. A deep embedded dimensionality reduction clustering (DERC) model [31] was designed with a probability-based triplet loss to improve the clustering accuracy by combining embedded features and dimensionality reduction into the image clustering process. Partition confidence maximization (PICA) [33] was developed to find the most confident cluster decision boundary by learning the most confident clusters from all possible solutions and to determine the most semantically possible class separation. However, its performance is unstable because it only addresses the semantic distribution at the instance-cluster level. Subsequently, a clustering model with prior information [42] was suggested to concurrently learn hierarchical representations and cluster assignments jointly by minimizing the discrepancy between each pair of instance assignments, where different distance metrics for each data point were examined. Recently, deep robust clustering (DRC) [19] was developed to investigate deep clustering from two perspectives: assignment probability and assignment features. The loss function was designed by maximizing the mutual information between the cluster assignment distribution of the images and their augmentations. A similar mutual information-based contrastive loss was employed in CRLC [41], where a weighted sum of two instance-level contrastive losses with respect to the feature representation head and clustering head was used to maximize mutual information across various augmentations. Contrastive clustering (CC) [21] defined an objective function consisting of instance-level and cluster-level losses to design an online image clustering algorithm by maximizing the similarities of one image and its corresponding augmentations and minimizing those of negative ones. Prototypical contrastive learning (PCL) [40] formulated prototypical contrastive learning as an expectation–maximization algorithm to train a deep CNN and perform iterative clustering and representation learning in an EM-based framework. However, these methods suffer from cluster collision because the generated pseudo-positive instances may not be truly positive. In [43], a clustering framework was trained by combining instance-level contrastive learning with cluster center prediction, which employed an ensemble objective loss function that combined instance-level contrastive loss, KL-divergence of clustering loss, and an aggregated anchor function. Instead of comparing feature embeddings directly in the latent space, a clustering-based representation algorithm [36] utilized a swapped prediction mechanism for image clustering without pairwise comparisons. Cross-entropy loss was used to measure the divergence between the generated code and the cluster-assigning vectors.

Joint-training-based clustering models have achieved significant improvements compared with alternating-training-based methods by avoiding error accumulation. However, they have limited ability to discriminate the semantic cluster because the intra-class boundaries are neither modeled by the cluster assignment derived from instance-level features nor captured by generating pseudo-labels at the instance level or by employing data augmentations. To model semantic-based decision boundaries, DCSC was proposed in this study to learn cluster semantic structures using a three-pronged objective function that is reinforced by semantic consistency and cluster-level discrimination. DCSC implicitly models inter-cluster and intra-cluster decision boundaries. DCSC generates a larger margin between clusters and a smaller distance within the same clusters compared to that of other SOTA methods.

3. Deep clustering model based on semantic consistency (DCSC) model

Given a dataset containing N unlabeled images $\mathfrak{T}= \{I_1, I_2, \ldots, I_N\}$, deep clustering aims to group the images into K clusters to ensure that the images within the same cluster are close, whereas the images in different clusters are separated. An improved deep clustering model (DCSC) was proposed to learn the semantic cluster boundaries. Fig. 1 shows the overall framework. The proposed DCSC model consists of four components: (1) a deep neural network $f_\theta(\cdot)$, which is used as a backbone encoder to learn the feature representation; (2) a feature-level projection head *Fproj*, which is employed to derive the embedded features; (3) a cluster-level projection head *Cproj*, which is leveraged to estimate the cluster assignment at the cluster-level; and (4) a semantic-level swapped prediction head consisting of a *Sproj* and a *Spred*, which is used to learn the semantic distribution of instances.

3.1. Objective function of the DCSC

The performance of a contrastive learning-based clustering algorithm significantly depends on its objective function. The objective function of DCSC consists of four components: (1) a feature-level contrastive loss (FLC) with respect to pairs of instances in latent space, to pull positive pairs closer and push negative sample pairs apart; (2) a cluster-level contrastive loss (CLC) of cluster assignments for all instances to find a conservative decision boundary to maximize the distance between clusters; (3) a semantic-level contrastive loss (SLC) to learn the semantic distribution of instances; and 4) a clustering regularization term (CR), to avoid the local minimum caused by clustering a majority of instances into a minority of clusters.

♦ **Feature-level contrastive loss**

Given an unlabeled dataset $\mathfrak{T}$ with N samples, we performed two types of random data augmentation methods on each image and obtained $2N$ augmented instances, denoted by $X^{Aug} = \left\{x_1^a, x_2^a, \ldots, x_N^a, x_1^b, x_2^b, \ldots, x_N^b\right\}$. Each pair of instances $\{x_i^a, x_j^b\}$ is labeled as positive samples when *i=j* and negative counterparts if $i \neq j$. Therefore, the feature-level contrastive (FLC) loss for the data augmentations of image x_i is expressed in Eq. (3.1).

$$fl_i^a{=}{-}log\frac{\exp(s(z_i^a, z_i^b)/\tau_{FLC})}{\sum_{j=1}^{N}[\exp(s(z_i^a, z_j^a)/\tau_{FLC}){+}\exp(s(z_i^a, z_j^b)/\tau_{FLC})]},\quad i, j = 1, 2, \ldots, N \tag{3.1}$$

where z_i^a is the feature embedding of the *i*th image with augmentation ***a***, and z_j^b is the corresponding feature embedding of x_j with augmentation ***b***. τ_{FLC} is a temperature parameter and $s(\cdot)$ is the similarity of two augmented data samples. As a result, the corresponding symmetric FLC loss is computed across all pairs in a mini-batch, as given by Eq. (3.2).

$$L_{FLC} = \frac{1}{2N}\sum_{i=1}^{N}(fl_i^a + fl_i^b) \tag{3.2}$$

The FLC loss is leveraged to maximize the similarities of the positive instances and minimize those of the negative instances at the feature level.

♦ **Cluster-level contrastive loss**

Given a collection of instances with a batch size of N and a potential class number of K, the purpose of the DCSC is to correctly predict the cluster assignment for each instance. In an ideal scenario, each instance should be assigned to only one cluster, which is the same as the ground-truth label in classification tasks. A cluster-level contrastive loss (CLC loss) was devised to encourage the cluster assignment distribution vectors of N images falling into different clusters to be orthogonal to enable the learning process towards this ideal case. Specifically, let $C^a \in R^{N\times K}$ and $C^b \in R^{N\times K}$ denote the cluster assignment matrix for N instances with two randomly selected augmentations, ***a*** and ***b***, respectively, where $C_{.j}^a$ is the *j*th cluster assignment distribution of all N instances with augmentation ***a*** in a mini-batch. Let C be the concatenation of C^a and C^b. The *j*th column of C should preferably be orthogonal to each other, except for the (*k+j*)th column. Therefore, a cluster structure discovering loss is defined in Eq. (3.3) to distinguish the *i*th cluster from other clusters.

$$L_{CLC} = -\frac{1}{2K}\sum_{j=1}^{K}\left[\log\frac{\exp(s(C_{\cdot,j}^a, C_{\cdot,j}^b)/\tau_{CLC})}{\sum_{k=1,k\neq j}^{2K}[\exp(s(C_{\cdot,j}, C_{\cdot,\mathrm{k}})/\tau_{CLC})]} + \log\frac{\exp(s(C_{\cdot,j}^b, C_{\cdot,j}^a)/\tau_{CLC})}{\sum_{k=1,\mathrm{k}\neq j}^{2K}[\exp(s(C_{\cdot,j}, C_{\cdot,\mathrm{k}})/\tau_{CLC})]}\right], \tag{3.3}$$

where $C_{.j}^a$ and $C_{.j}^b$ are the *j*th columns of C^a and C^b, representing the cluster distribution of the *j*th cluster over all N instances with augmentations ***a*** and ***b***, respectively. τ_{CLC} is a temperature parameter, where K is a predefined number of clusters.

The CLC loss was utilized to distinguish the cluster distribution generated by the guaranteed positive examples from other distributions (i.e., the CLC loss attempts to find a conservative decision boundary that maximizes the distance between clusters).

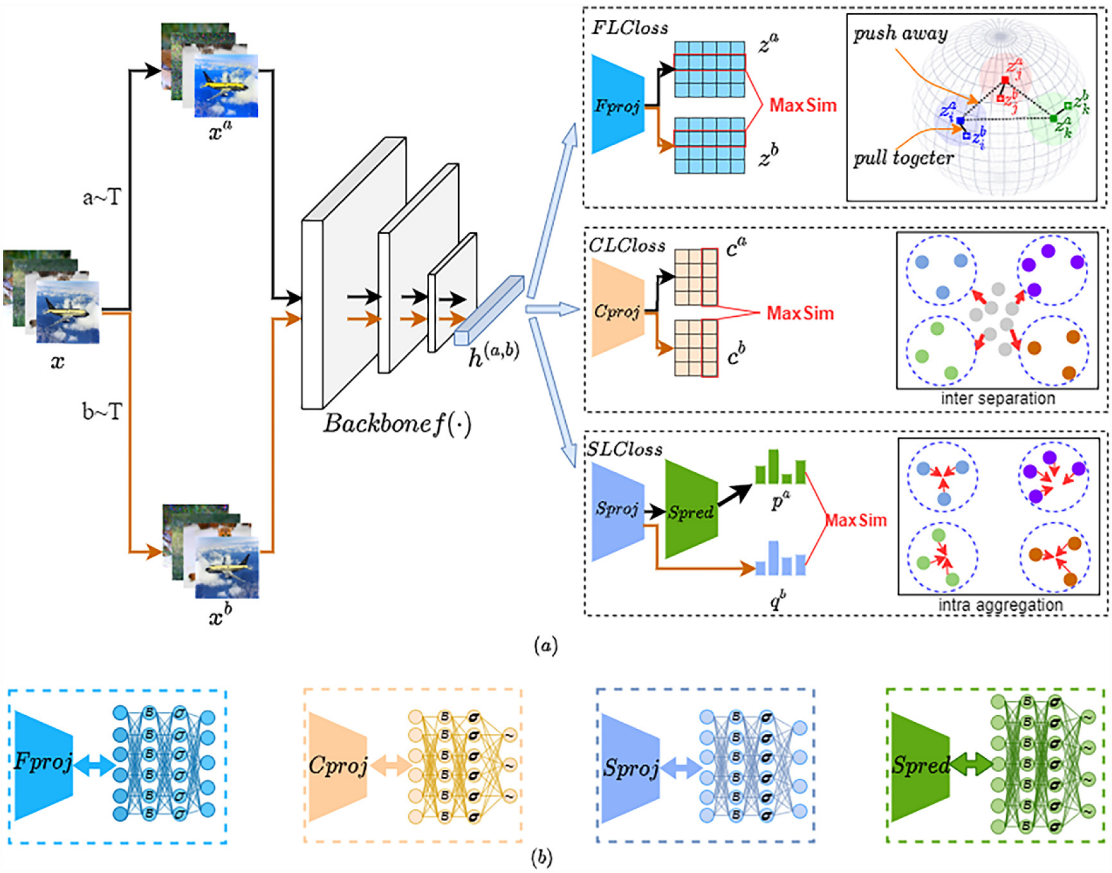


Fig. 1. (a) An overall framework of the improved image clustering model based on semantic contrastive learning, where two loss heads (CLC and SLC losses) are added to encourage the model to learn more semantic cluster boundaries; (b) Structure of three separate non-linear projection heads and one prediction head, where B denotes the batch normalization layer, σ denotes the ReLU activation layer, and $\sim$ denotes the Soft-max function.

♦ Semantic-level contrastive loss

A semantic-level contrastive loss is used to guide the backbone encoder to learn the semantic representations of the instances. In classification and semantic segmentation settings, labels can be regarded as high-level semantic representations. Therefore, we encoded all input instances into a latent space with the dimensionality of the cluster number and regarded their soft labels as a special semantic representation. An asymmetric network was introduced to derive the distribution of instances semantically in DCSC because an asymmetric structure can avoid the high intra-class diversities caused by the sensitivity to the maximum value of the sof-tmax function [47].

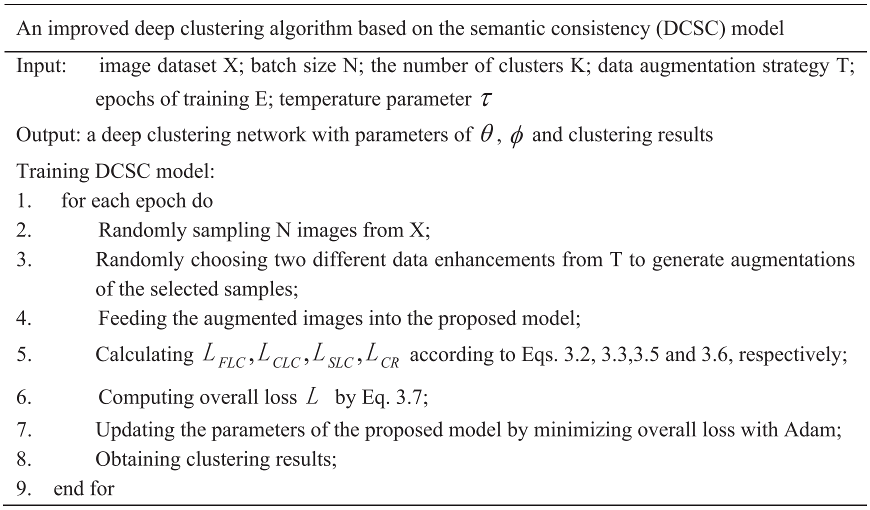
Given an unlabeled image $I \in \mathfrak{T}$, the first step is to compute the cluster assignment distribution of its augmented versions in the feature embedding space using an asymmetric network (i.e., $p_\theta(P|h_i^a)$ and $q_{\theta'}(Q|h_i^a)$ are obtained, where θ and θ' are the parameters of the network and P and Q are the cluster assignment of N images). The second step is to minimize the divergence between the two distributions of $p_\theta(P|h_i^a)$ and $q_{\theta'}(Q|h_i^a)$. Specifically, two different augmented versions of the same instance were fed to an encoder together with an asymmetric prediction MLP layer, and their corresponding embedded features were transformed into their semantic distributions by a prediction head or soft-max layer separately, yielding two distributions of $p_\theta(P|h_i^a)$ and $q_{\theta'}(Q|h_i^a)$. Without the loss of generality, the contrastive loss between the probability distributions of the ith instance at the semantic level is defined in Eq. (3.4).

$$sl_i^a = -\log \frac{\exp(s(p_i^a, q_i^b)/\tau_{SLC})}{\sum_{j=1}^{N}[\exp(s(p_i^a, q_j^a)/\tau_{SLC}) + \exp(s(p_j^a, q_i^b)/\tau_{SLC})]} \qquad (3.4)$$

where $s(p_i^a, q_j^b)$ denotes the similarity between the probability distributions of the cluster assignment of the ith and jth instances, $i, j = 1, 2, \ldots, N$, which can be defined as the Jensen–Shannon divergence or KL divergence of the two distributions. In this study, τ_{SLC} is a temperature parameter that takes a higher value to encourage more compactness within the cluster. A larger value of τ_{SLC} tends to penalize less on the nearest neighbor samples, keeping semantically similar instances in the same clusters. The semantic-level contrastive loss is obtained using Eq. (3.5) by traversing all the instances in the batch.

$$\mathrm{L}_{SLC} = \frac{1}{2N}\sum_{i=1}^{N}(sl_i^a + sl_i^b) \qquad (3.5)$$

The semantic-level contrastive loss is leveraged to differentiate the probability distribution of the cluster assignment generated by the guaranteed positive instances from other instances at the semantic level to find a conservative decision boundary. It was used to minimize the distance between positive samples and maximize the distance between negative samples. In addition, instances with different ground-truth labels may be clustered into the same cluster, resulting in a cluster collision problem because the soft-max function is only sensitive to the maximum value of the outputs. This problem is solved using an asymmetric Siamese network with two branches of non-linear projection heads, where one branch comprises a projector and a predictor, and the other has a non-linear projection head only. The experimental results in Section 4.4.2 show that the parameters of the two branches can be adjusted during the training process.



An improved deep clustering algorithm based on the semantic consistency (DCSC) model

```
Input:   image dataset X; batch size N; the number of clusters K; data augmentation strategy T;
         epochs of training E; temperature parameter $\tau$
Output: a deep clustering network with parameters of $\theta$, $\phi$ and clustering results
Training DCSC model:
for each epoch do
    Randomly sampling N images from X;
    Randomly choosing two different data enhancements from T to generate augmentations
    of the selected samples;
    Feeding the augmented images into the proposed model;
    Calculating $L_{FLC}, L_{CLC}, L_{SLC}, L_{CR}$ according to Eqs. 3.2, 3.3,3.5 and 3.6, respectively;
    Computing overall loss $L$ by Eq. 3.7;
    Updating the parameters of the proposed model by minimizing overall loss with Adam;
    Obtaining clustering results;
end for
```

♦ **Clustering regularization term**

An entropy constraint was applied as a clustering regularization term to solve this problem (see Eq. (3.6)) to avoid the local optimal solution produced by assigning the majority of examples to a minority of clusters during training.

$$\mathrm{L}_{CR} = -[p(c^a)\log p(c^a) + p(c^b)\log p(c^b)], \tag{3.6}$$

where $p(c^t) = \frac{\sum_{i=1}^{N} c_{ij}^t}{N}$ and $t \in \{a, b\}$ is the probability distribution of the jth cluster for all instances with augmentation $\boldsymbol{t}$ in the entire batch, $j = 1,2,\ldots, K$ and $t \in \{a, b\}$. In summary, the final objective function of the DCSC is defined by Eqs. (3.7), where λ is the balance parameter.

$$L = L_{FLC} + L_{CLC} + L_{SLC} + \lambda L_{CR} \tag{3.7}$$

3.2. Algorithm of deep clustering model based on semantic consistency

The proposed clustering method based on semantic consistency is summarized in the following Algorithm.

4. Experiments

In this section, various comparison experiments were conducted on widely used benchmark datasets to validate the performance of the DCSC. Several ablation studies have been performed to investigate the importance of each loss item and its effect on each DCSC module.

4.1. Datasets and evaluation metrics

Six widely used benchmark datasets, CIFAR-10/100 [48], STL-10 [49], ImageNet-10 [14], ImageNet-Dogs [14] and Tiny-ImageNet [50], were used in this study. The number of samples and cluster number ranged from 500 to 100,000 and from 10 to 200. We trained the model using the instances in each dataset and evaluated its clustering performance on the same dataset to validate the performance of the DCSC on different datasets.

Three metrics were employed to evaluate the clustering performance of the DCSC, including accuracy (ACC) [51], normalized mutual information (NMI) [52] and adjusted rand index (ARI) [53]. The higher the value of these metrics, the better the clustering performance. We reported the average value of each metric for each dataset to reduce the stochasticity of the experimental results.

4.2. Experiments implementation details

Image augmentations For a fair comparison, the same augmentation strategy and backbone encoder used in [21,35] are adopted in this study. The data augmentations were randomly selected from *ResizedCrop*, *HorizontalFlip*, *ColorJitter*, *Grayscale*, or *GaussianBlur*, which have been proved more effective for representation learning algorithms [35].

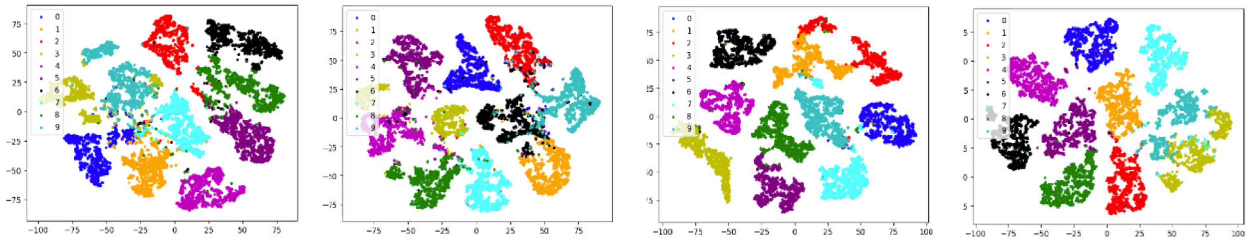
Backbone: To ensure a fair comparison, ResNet34 [19,21,35, 54] was adopted as the backbone encoder, although DCSC does not depend on any specific network. As ResNet was designed for images of size 224 × 224 pixels, all input images were resized before they were supplied to the backbone encoder. Each image was first resized to 224 × 224 to ensure that the model can applied to every dataset to keep the structure of the backbone encoder unchanged. In non-linear projection heads, batch normalization (BN) [55] was used to accelerate the training process and increase clustering efficiency. The projection head and prediction head are jointly trained in a swapped manner based on the semantic-level loss to solve the cluster collision induced by utilizing the softmax function exclusively in the non-linear prediction head. In this study, the backbone encoder used was not pre-trained on any dataset.

Parameters: The embedded feature dimension was set to 128 to make a fair comparison with previous studies. For the feature-level contrastive loss and cluster-level contrastive loss, the temperature parameters τ_{FLC} and τ_{CLC} were set as 0.2 and 1.0, respectively. For SLC loss, τ_{SLC} is set to 1.0 to avoid cluster collisions. In addition, an over-clustering approach was used to extract additional information from potentially relevant classes, reducing the possibility of generating an overly conservative boundary using only positive instances. Specifically, in our experiments, we set the number of clusters for Tiny-ImageNet to 700 and 70 for the other datasets in our study. The balance parameter λ was set to 1 in all experiments.

Optimizer: We implemented DCSC using Pytorch 1.6.0. We used the Adam optimizer [56] with a learning rate of 3e-4 for gradient descent during the training procedure. Because of memory limitations, the batch size and training epochs were set to 256 and 1,000, respectively. All experiments were performed using an Nvidia 3090 RTX 24G. CIFAR-10 requires 55 GPU hours, CIFAR-100 requires 60 GPU hours, STL-10 requires 120 GPU hours, ImageNet-10 requires 15 GPU hours, ImageNet-dogs requires 20 GPU hours, and Tiny-ImageNet requires 100 GPU hours to train the proposed DCSC.

Table 1
Comparison results on six image clustering benchmarks.

Dataset	CIFAR-10			CIFAR-100			STL-10			ImageNet-10			ImageNet-Dogs			Tiny-ImageNet		
Metrics	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means	0.229	0.087	0.049	0.130	0.084	0.028	0.192	0.125	0.061	0.241	0.119	0.057	0.105	0.055	0.020	0.025	0.065	0.005
SC	0.247	0.103	0.085	0.136	0.090	0.022	0.159	0.098	0.048	0.274	0.151	0.076	0.111	0.038	0.013	0.022	0.063	0.004
AC	0.228	0.105	0.066	0.138	0.098	0.034	0.332	0.239	0.140	0.242	0.138	0.067	0.139	0.037	0.021	0.027	0.069	0.005
NMF	0.190	0.081	0.034	0.118	0.079	0.026	0.180	0.096	0.046	0.230	0.132	0.065	0.118	0.044	0.016	0.029	0.072	0.005
DCGAN	0.315	0.265	0.176	0.151	0.120	0.045	0.298	0.210	0.139	0.346	0.225	0.157	0.174	0.121	0.078	0.041	0.135	0.007
DeCNN	0.282	0.240	0.174	0.133	0.092	0.038	0.299	0.227	0.162	0.313	0.186	0.142	0.175	0.098	0.073	0.036	0.111	0.006
AE	0.314	0.239	0.169	0.165	0.100	0.048	0.303	0.250	0.161	0.317	0.210	0.152	0.185	0.104	0.073	0.041	0.131	0.007
VAE	0.291	0.245	0.167	0.152	0.108	0.040	0.282	0.200	0.146	0.334	0.193	0.168	0.179	0.107	0.079	0.033	0.113	0.006
JULE	0.272	0.192	0.138	0.137	0.103	0.033	0.277	0.182	0.164	0.300	0.175	0.138	0.138	0.054	0.028	0.033	0.102	0.006
DEC	0.301	0.257	0.161	0.185	0.136	0.050	0.359	0.276	0.186	0.381	0.282	0.203	0.195	0.122	0.079	0.037	0.115	0.007
DC-VAE	0.350	0.440	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DAC	0.522	0.396	0.306	0.238	0.185	0.088	0.470	0.366	0.257	0.527	0.394	0.302	0.275	0.219	0.111	0.066	0.190	0.017
DDC	0.524	0.424	0.329	-	-	-	0.489	0.371	0.267	0.577	0.433	0.345	-	-	-	-	-	-
DCCM	0.623	0.496	0.408	0.327	0.285	0.173	0.482	0.376	0.262	0.710	0.608	0.555	0.383	0.321	0.182	0.108	0.224	0.038
PICA	0.696	0.591	0.512	0.337	0.310	0.171	0.713	0.611	0.531	0.870	0.802	0.761	0.352	0.352	0.201	0.098	0.277	0.040
DRC	0.727	0.621	0.547	0.367	0.356	0.208	0.747	0.644	0.569	0.884	0.830	0.798	0.389	0.384	0.233	0.139	0.321	0.056
CRLC	**0.799**	0.679	0.634	0.425	0.416	0.263	0.818	0.729	0.682	0.854	0.831	0.759	0.461	0.484	0.297	-	-	-
CC	0.790	**0.705**	0.637	0.429	0.431	0.266	0.850	0.764	0.726	0.893	0.859	0.822	0.429	0.445	0.274	0.140	0.340	0.071
DCSC	0.798	0.704	**0.644**	**0.469**	**0.452**	**0.293**	**0.865**	**0.792**	**0.749**	**0.904**	**0.867**	**0.838**	**0.443**	**0.462**	**0.299**	**0.171**	**0.358**	**0.073**



(a) PICA (b) DRC (c) CC (d) DCSC

Fig. 2. Visualization of clustering result of four different methods on ImageNet-10.

4.3. Comparisons with state-of-the-art methods

The performance of the DCSC was comprehensively evaluated using six benchmark datasets. The results were compared with 18 image clustering methods, including typical traditional clustering models, such as K-means [5], SC [6], AC [7], and NMF [22]; and popular deep learning-based clustering models, such as DCGAN [11], DeCNN [10], AE [16], DC-VAE [53], VAE [57], JULE [23], DEC [12], DAC [14], DDC [24], DCCM [25], PICA [33], DRC [19], CRLC [41], and CC [21]. The results of the existing methods were obtained from [19,25,33], and the final comparison results are summarized in Table 1, where the best results are shown in bold.

Table 1 shows that the proposed DCSC model performed better than the existing models on the six benchmark datasets. For example, compared with the result of CC [21], the accuracy of the clustering results provided by DCSC is improved by 9.3% and 22.1% on CIFAR-100 and Tiny-ImageNet, respectively.

Fig. 2 shows the results of the comparison of the proposed model with three SOTA models (i.e., PICA [33], DRC [19], and CC [21]). The proposed DCSC generates smaller intracluster divergence compared to the other SOTA methods, as shown in Fig. 2.

4.4. Ablation studies

In this section, various ablation studies are conducted on two benchmark datasets. This aims to (1) investigate the importance of different losses in the proposed objective function, (2) disclose the effect of each model component on the DCSC model, and (3) explore the functions of batch normalization (BN) and asymmetric networks in contrastive learning-based clustering models.

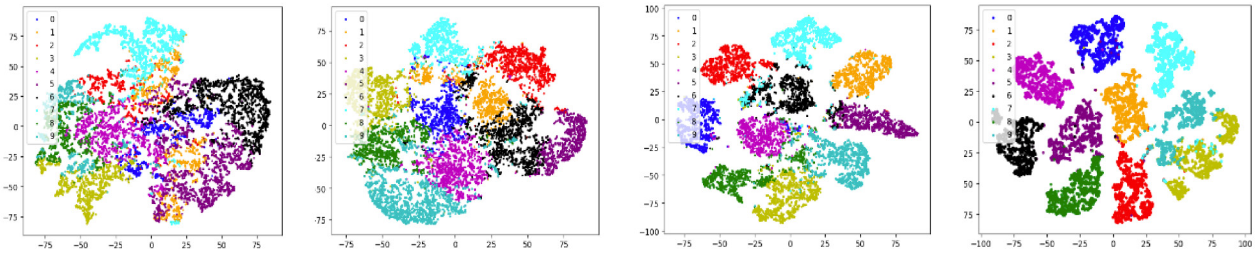
Table 2
Experimental results of different losses.

Dataset	Metric	NMI	ACC	ARI
CIFAR-10	DCSC	**0.704**	**0.798**	**0.644**
	DCSC *w/o* FLC loss	0.611	0.694	0.523
	DCSC *w/o* CLC loss	0.673	0.761	0.625
	DCSC *w/o* SLC loss	0.657	0.745	0.578
ImageNet-10	DCSC	**0.867**	**0.904**	**0.838**
	DCSC *w/o* FLC loss	0.857	0.895	0.824
	DCSC *w/o* CLC loss	0.851	0.890	0.819
	DCSC *w/o* SLC loss	0.860	0.896	0.831

4.4.1. Importance of different losses

We conducted ablation experiments on CIFAR-10 and ImageNet-10 by removing different losses from the original objective function to explore the importance of different losses in the DCSC. Table 2 lists the ablation results.

For a fair comparison with CC [21], K-means is performed for clustering in the embedding feature space with the same dimension after removing the CLC/SLC loss. For CIFAR-10 and ImageNet-10, the proposed DCSC model with three contrastive losses (three-head) produced the best results, as shown in Table 2. Table 2 shows that each loss item contributes to the enhancement of the clustering performance. For example, the removal of the semantic-level contrastive loss reduced the performance by 7.1% in terms of ACC and 7.2% in terms of NMI on CIFAR-10. Deleting



a) 0 epoch (ACC=0.306) b) 30 epochs (ACC=0.632) c) 200 epochs (ACC=0.849) d) 1000 epochs (ACC=0.904)

Fig. 3. Predicted dynamics of DCSC during the training process on ImageNet-10.

Table 3
Effect of BN in MLP.

Datasets	BN in MLP?	ACC	NMI	ARI
CIFAR10	No	0.796	0.702	0.641
	Yes	**0.798**	**0.704**	**0.644**
CIFAR100	No	0.463	0.488	0.289
	Yes	**0.469**	**0.452**	**0.293**

Table 4
Effect of predictive head.

Dataset/Metric	CIFAR-100			ImageNet-10		
	ACC	NMI	ARI	ACC	NMI	ARI
DCSC	**0.469**	**0.452**	**0.293**	**0.904**	**0.867**	**0.838**
DCSC *w/o Spred* head	0.449	0.440	0.278	0.890	0.861	0.824

the cluster-level contrastive loss decreased ACC and NMI by 1.8% and 1.6%, respectively, on ImageNet-10. The results in Table 2 indicate that the joint training of multiheads improves the semantic representation ability of the backbone encoder. For example, in CIFAR-10, jointly training CLC and SLC in DCSC achieves NMI and ACC of 61.1% and 69.4%, respectively. Only the training of CLC loss in the CC model [21] achieves NMI and ACC of 59.2% and 65.9% (see Table 4 in [21]) respectively. Table 2 shows that SLC loss is more important than CLC loss in the CIFAR-10 dataset. In contrast, the CLC loss mainly contributed on the performance improvement on the ImageNet-10 dataset, with ACC and NMI of 1.5% and 1.8%, respectively.

4.4.2. Impact of batch normalization and asymmetric network on DCSC

Learning a good image representation is key for efficient training of downstream tasks. Most contrastive learning-based representation learning methods rely on either large batch sizes [35,58] or memory banks [34] to improve the diversity of negative samples. BYOL [44] provides an alternative way to train a backbone encoder by adding a BN layer and using a predictor network that breaks network symmetry. These methods can successfully avoid trivial solutions while allowing smaller batch sizes without sacrificing the performance. Some researchers claimed that BN played an important role in preventing collapse [44,59]. Other research [60,61] asserted that the asymmetry of networks was more important than BN. We performed ablation studies to investigate the impact of the BN layer and asymmetry of networks on the clustering performance of the DCSC, and the experimental results are shown in Table 3.

Table 3 shows that the application of BN before the activation function in non-linear projection MLP heads slightly improves clustering performance, especially for fine-grained semantic information extraction tasks. This is because the BN layer can render the backbone encoder more robust when the initialization is improperly scaled. Additionally, the BN layer can maintain a consistent data distribution by constraining the latent feature space to a unit hypersphere, thereby speeding up the training process.

We performed ablation experiments using CIFAR-100 and ImageNet-10 to explore the effect of the asymmetric structure in the DCSC model. The experimental results are summarized in Table 4.

Based on Tables 3 and 4, the removal of BN from the DCSC degrades the performance by 1.27%, whereas the deletion of the predictive head reduces the accuracy by 4.26%. The empirical results in Tables 3 and 4 suggest that in DCSC, the asymmetric network is more essential than BN. This conclusion is consistent with the results of Simsiam [60] (see Table 1 in [60] for more details). Therefore, we concluded that introducing asymmetry in the architecture plays a more important role than BN in aiding the backbone encoder to avoid trivial solutions. One reason for this is the symmetric loss expressed in Eqs. (3.4) and (3.5) [60]. Another possible reason is that the asymmetric structure can jointly train the projection and prediction heads simultaneously, thereby avoiding the generation of trivial solutions.

4.5. Qualitative study

4.5.1. Visualization of clustering result

In this section, the evolution of cluster assignment during the training process is visualized using t-SNE [62]. A qualitative analysis was presented to better understand the DCSC model. Fig. 3 shows the results of the DCSC on ImageNet-10, where different colors represent different clusters. A total of 13,000 samples were used, where the ground-truth class labels are shown in different colors. Fig. 3 shows that the distance between clusters increased and the samples within the same cluster became more compact as the training epochs increased.

4.5.2. Investigation of success and failure cases

Examining the success and failure cases of the DCSC model will provide more insights into our method. In this section, we randomly selected four classes of images from ImageNet-10 and focused on the three cases. Fig. 4(left) shows successful cases, in which each image is correctly assigned to its ground-truth label. Fig. 4(middle) presented false negative failure cases, in which images from a target class are incorrectly clustered to other clusters. Finally, Fig. 4 (right) illustrates false positive failure cases, in which images from other classes are incorrectly assigned to the current target class (right).

Fig. 4 shows that the DCSC model can successfully cluster images in most scenarios, indicating that the proposed clustering model is valid. However, misclustering outcomes occur when the DCSC is used to discriminate samples with high similarity. Taking false negative examples as an example, most misclustered samples have more than one ground-truth label, which could worsen

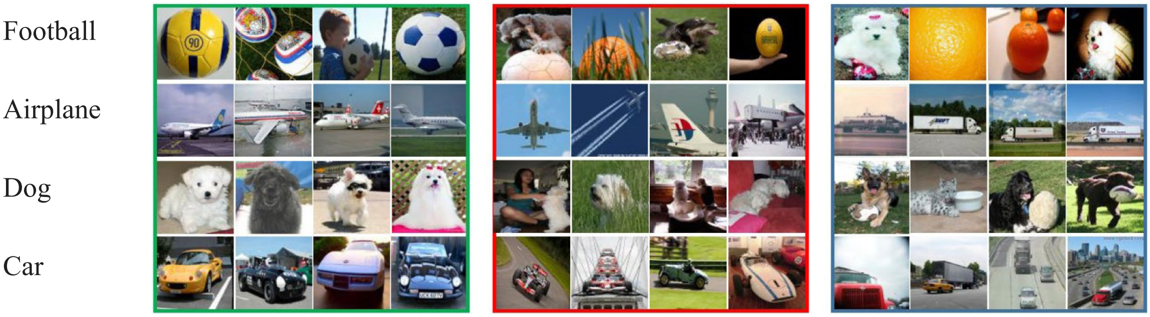


Fig. 4. Success vs. failure cases studies on ImageNet-10. (Left) Successful positive cases, (Middle) false negative, and (right) false positive failure cases.

the performance of the proposed model. One possible explanation for these failures is that learning appropriate feature embeddings without ground-truth labels is difficult for these models.

5. Conclusions and future works

DCSC was proposed in this study. DCSC imposes the cluster structure discovery on the learning process and enhances the performance of existing contrastive learning-based approaches, increasing inter-class diversity and decreasing intra-class diversity by effectively integrating the semantic clustering assignment distribution loss into the traditional contrastive loss. Extensive experimental results demonstrate that the proposed DCSC outperforms the state-of-the-art image clustering methods. However, some misclustering results are unavoidable because of a lack of supervision information when DCSC is used to differentiate fine-grain clusters.

For future research, we will focus on how to leverage pseudo-labels with high confidence to improve the ability to distinguish fine-grained classes. Another possible direction is to investigate the relationship of the embedding features to reduce redundant features and generate a more efficient representation of each input.

CRediT authorship contribution statement

Feng Zhang: Conceptualization, Methodology, Writing – original draft. **Lin Li:** Experiments, Validation, Writing assistant. **Qiang Hua:** Resources, Funding acquisition, Editing. **Chun-Ru Dong:** Revision, Editing. **Boon-Han Lim:** Editing.

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.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable time. This research is partly supported by the National Nature Science Fund Project NSFCs, China (No. 61773150), the Natural Science Foundation of Hebei Province, China (No. F2018201115), and the Key Scientific Research Foundation of Education Department of Hebei Province, China (ZD2019021).

References

[1] S. Chakraborty, S. Das, Detecting meaningful clusters from high-dimensional data: A strongly consistent sparse center-based clustering approach, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[2] 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, AAAI'19, vol. 33, no. 01, 2019, pp. 8134–8141.

[3] X. Zhang, Y. Sun, H. Liu, et al., Improved clustering algorithms for image segmentation based on non-local information and back projection, Inform. Sci. 550 (2021) 129–144.

[4] J. Wu, K. Long, F. Wang, et al., Deep comprehensive correlation mining for image clustering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR'19, 2019, pp. 8150–8159.

[5] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 14, 1967, pp. 281–297.

[6] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, NIPS'02, 2002, pp. 849–856.

[7] K.C. Gowda, G. Krishna, Agglomerative clustering using the concept of mutual nearest neighbourhood, Pattern Recognit. 10 (2) (1978) 105–112.

[8] D.G. Lowe, Object recognition from local scale-invariant features, 1999, pp. 1150–1157,

[9] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, vol. 1, 2005, pp. 886–893.

[10] M.D. Zeiler, D. Krishnan, G.W. Taylor, et al., Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'10, 2010, pp. 2528–2535.

[11] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, Comput. Sci. (2015).

[12] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International Conference on Machine Learning, ICML'16, PMLR, 2016, pp. 478–487.

[13] F. Li, H. Qiao, B. Zhang, Discriminatively boosted image clustering with fully convolutional auto-encoders, Pattern Recognit. 83 (2018) 161–173.

[14] J. Chang, L. Wang, G. Meng, et al., Deep adaptive image clustering, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV'17, 2017, pp. 5879–5887.

[15] J. Zhang, C.G. Li, C. You, et al., Self-supervised convolutional subspace clustering network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR'19, 2019, pp. 5473–5482.

[16] K. Ghasedi Dizaji, A. Herandi, C. Deng, et al., Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV'17, 2017, pp. 5736–5745.

[17] P. Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'18, 2018, pp. 1596–1604.

[18] X. Ji, J.F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR'19, 2019, pp. 9865–9874.

[19] H. Zhong, C. Chen, Z. Jin, et al., Deep robust clustering by contrastive learning, 2020, CoRR arXiv:abs/2008.03030.

[20] Z. Dang, C. Deng, X. Yang, et al., Doubly contrastive deep clustering, 2021, arXiv preprint arXiv:2103.05484.

[21] Y. Li, P. Hu, Z. Liu, et al., Contrastive clustering, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI'21, 2–9 Feb 2021, Vancouver, Canada.

[22] D. Cai, X. He, X. Wang, et al., Locality preserving nonnegative matrix factorization, in: Proceedings of the 21th International Joint Conference on Artificial Intelligence, 2009.

[23] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'16, 2016, pp. 5147–5156.

[24] J. Chang, Y. Guo, L. Wang, et al., Deep discriminative clustering analysis, 2019, arXiv preprint arXiv:1905.01681.

[25] J. Wu, K. Long, F. Wang, et al., Deep comprehensive correlation mining for image clustering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR'19, 2019, pp. 8150–8159.

[26] P. Haeusser, J. Plapp, V. Golkov, et al., Associative deep clustering: Training a classification network with no labels, in: German Conference on Pattern Recognition, Springer, Cham, 2018, pp. 18–32.

[27] B. Yang, X. Fu, N.D. Sidiropoulos, et al., Towards k-means-friendly spaces: Simultaneous deep learning and clustering, in: International Conference on Machine Learning, ICLR'17, 2017, pp. 3861–3870.

[28] Deep. Clustering, M. Caron, P. Bojanowski, A. Joulin, et al., Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, ECCV'18, 2018, pp. 132–149.

[29] Z. Chen, X. Zhang, W. Pedrycz, et al., K -means clustering for the aggregation of HFLTS possibility distributions: N -two-stage algorithmic paradigm, Knowl.-Based Syst. 227 (2021) 107230.

[30] R.T.Q. Chen, X. Li, R. Grosse, et al., Isolating sources of disentanglement in variational autoencoders, Adv. Neural Inf. Process. Syst. (2018) 31.

[31] Y. Yan, H. Hao, B. Xu, et al., Image clustering via deep embedded dimensionality reduction and probability-based triplet loss, IEEE Trans. Image Process. 29 (2020) 5652–5661.

[32] H. Jia, Y.M. Cheung, Subspace clustering of categorical and numerical data with an unknown number of clusters, IEEE Trans. Neural Netw. Learn. Syst. 29 (8) (2017) 3308–3325.

[33] J. Huang, S. Gong, X. Zhu, Deep semantic clustering by partition confidence maximization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR'20, 2020, pp. 8849–8858.

[34] K. He, H. Fan, Y. Wu, et al., Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR'20, 2020, pp. 9729–9738.

[35] T. Chen, S. Kornblith, M. Norouzi, et al., A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, ICML'20, PMLR, 2020, pp. 1597–1607.

[36] M. Caron, I. Misra, J. Mairal, et al., Unsupervised learning of visual features by contrasting cluster assignments, in: Advances in Neural Information Processing Systems, NIPS'20, vol. 33, 2020, pp. 9912–9924.

[37] S. Becker, G.E. Hinton, Self-organizing neural network that discovers surfaces in random-dot stereograms, Nature 355 (6356) (1992) 161–163.

[38] S. Purushwalkam, A. Gupta, Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases, 2020, arXiv preprint arXiv:2007.13916.

[39] C. Niu, H. Shan, G. Wang, Spice: Semantic pseudo-labeling for image clustering, 2021, arXiv preprint arXiv:2103.09382.

[40] J. Li, P. Zhou, C. Xiong, et al., Pototypical contrastive learning of unsupervised representations, in: Proceedings of International Conference on Learning Representations, ICLR'21, 2021.

[41] K. Do, T. Tran, S. Venkatesh, Clustering by maximizing mutual information across views, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV'21, 2021, pp. 9928–9938.

[42] C. Ye, G. Gui, T. Ohtsuki, Deep clustering with lstm for vital signs separation in contact-free heart rate estimation, in: IEEE International Conference on Communications, ICC'20, IEEE, 2020, pp. 1–6.

[43] R. Sundareswaran, J. Herrera-Gerena, J. Just, et al., Cluster analysis with deep embeddings and contrastive learning, 2021, arXiv preprint arXiv:2109.12714.

[44] J.B. Grill, F. Strub, F. Altché, et al., Bootstrap your own latent - a new approach to self-supervised learning, Adv. Neural Inf. Process. Syst. 33 (2020) 21271–21284.

[45] K. Ohri, M. Kumar, Review on self-supervised image recognition using deep neural networks, Knowl.-Based Syst. 224 (2021) 107090.

[46] J. Huang, S. Gong, Deep clustering by semantic contrastive learning, 2021, arXiv preprint arXiv:2103.02662.

[47] Z. Huang, J. Chen, J. Zhang, et al., Exploring non-contrastive representation learning for deep clustering, 2021, arXiv preprint arXiv:2111.11821.

[48] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, 2009.

[49] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[50] Y. Le, X. Yang, Tiny ImageNet visual recognition challenge, CS 231N 7 (7) (2015) 3.

[51] S. Chang, J. Hu, T. Li, et al., Multi-view clustering via deep concept factorization, Knowl.-Based Syst. 217 (2) (2021) 106807.

[52] Y. Huang, Z. Shen, F. Cai, et al., Adaptive graph-based generalized regression model for unsupervised feature selection, Knowl.-Based Syst. 227 (2021) 107156.

[53] S. Rodriguez, F. Carvalho, Soft subspace clustering of interval-valued data with regularizations, Knowl.-Based Syst. 227 (2021) 107191.

[54] K. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'16, 2016, pp. 770–778.

[55] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, ICML'15, 2015, pp. 448–456.

[56] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 5rd International Conference for Learning Representations, ICLR'15, San Diego, 2015.

[57] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: Proceedings of the 4th International Conference for Learning Representations, ICLR'14, 2014.

[58] Y. Tian, C. Sun, B. Poole, et al., What makes for good views for contrastive learning, 2020, arXiv preprint arXiv:2005.10243.

[59] Y. Tian, Yu, L, Chen, X, et al., Understanding self-supervised learning with dual deep networks, 2020, arXiv preprint arXiv:2010.00578.

[60] X. Chen, K. He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR'21, 2021, pp. 15750–15758.

[61] P.H. Richemond, J.B. Grill, F. Altché, et al., BYOL works even without batch statistics, 2020, arXiv preprint arXiv:2010.10241.

[62] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008) 2579–2605.