Deep Self-supervised Subspace Clustering with Triple Loss

Xiaotong Bu[1,2], Jiwen Dong[1,2], Mengjiao Zhang[1,2], Guang Feng[1,2], Xizhan Gao[1,2], and Sijie Niu[1,2(✉)]

[1] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
sjniu@hotmail.com

[2] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, Jinan, China

Abstract. Deep subspace clustering (DSC) methods are widely used in various fields such as motion segmentation, image segmentation, and text mining. It uses the deep neural network to map high-dimensional features into low-dimensional latent subspace to achieve effective division of data. Nevertheless, DSC simply tends to learn representations based on auto-encoder, which can't fully exploit the intrinsic structure of the data. In this paper, we design a novel approach called Deep self-supervised subspace clustering with triple data (DSSCT), which aims to uncover supervised information inherent in the data. Specifically, DSSCT leverages data augmentation and triple contrastive loss to obtain more effective low-dimensional representations that capture the similarity and difference among different samples. In addition, we introduce a dual self-expression matrix fusion strategy to further enhance the discriminant of the self-expression matrix used in DSSCT. To evaluate the performance of our proposed method, we conduct extensive experiments on several widely used public datasets and achieved excellent performance when compared with other state-of-art methods.

Keywords: Deep subspace clustering · Triple contrastive loss · Dual self-expression matrix

1 Introduction

Clustering is an unsupervised learning methodology, which separates samples into corresponding classes without labels' information. With the development of the times, more and more complex high-dimensional data have emerged. However, traditional clustering algorithms such as k-means [11] and spectral clustering [5,8] have shown adverse effects in those data. Subspace clustering [10] assume that different subsets of features may exhibit distinct clustering patterns. Subspace clustering methods can be broadly categorized into four main categories, i.e., Iteration-based methods [1], Algebra-based methods [9], Statistical-based methods [28], and Self-expression-based methods [25]. Among them, self-expression-based subspace clustering has attracted a lot of researchers'

S. Rudinac et al. (Eds.): MMM 2024, LNCS 14555, pp. 423–436, 2024.
https://doi.org/10.1007/978-3-031-53308-2_31

attention. Numerous traditional subspace clustering methods has focused on improving subspace clustering performance by introducing various constraints on the matrix [8,25], but they have weak performance in dealing with non-linear subspace data. With the development of deep neural networks (DNNs), a few researchers proposed deep subspace clustering methods that employ neural networks to learn a non-linear mapping relationship between the original data and its latent feature [7,16,18]. However, conventional subspace clustering methods often rely on a single feature extraction, which may not capture the crucial features and exploit the rich information contained in data [13] e.g., light, darkness, shape, position, etc. These limitations undoubtedly impede the efficiency of subspace clustering.

Self-supervised learning allows for leveraging large amounts of unlabeled data to learn useful representations or features from the data. In recent years, Contrastive learning [4,27] has gained significant attention and success in self-supervised domains. It is based on the principle of encouraging the model to learn a distinction between similar samples and dissimilar samples. It allows the model to reduce the redundant features in learning representations and learn more discriminative features [12].

Motivated by recent progress in deep subspace clustering and self-supervised learning, in this paper we introduce a novel end-to-end trainable framework called DSSCT (Deep Self-Supervised Subspace Clustering with Triple Data). Our proposed method overcomes the limitation of traditional subspace clustering and shows excellent performance on benchmark dataset. The framework of the model is shown in Fig. 2. In our proposed method, we use the triple contrast module and the dual self-expression module for training to learn better low-dimensional latent features and enhance the discriminant of self-expression matrix. The main contributions of this paper are as follows:

(1) We propose a novel end-to-end trainable framework called DSSCT (Deep Self-Supervised Subspace Clustering with Triple Data), which employs a triple contrastive module to capture significant differences and extract self-expression information in latent subspace.
(2) We assume that the positive sample pairs lie in the same subspace and keep difference with negative samples. Based on this assumption, we design a dual self-expression structure and fuse two self-expression matrices as the affinity matrix to improve the discriminant ability of the affinity matrix.
(3) We conduct extensive experiments on four benchmark datasets to verify the superiority of our DSSCT.

2 Related Work

2.1 Deep Subspace Clustering

Subspace clustering is an effective clustering technique, which assumes data points whose lie in the same subspace can well be represented by a low-dimensional linear subspace [8]. Traditional subspace clustering for dealing with

linear subspace problems are based on spectral clustering [26]. They can be unified into two setps: a) building a affinity matrix; b) applying the matrix into spectral clustering. Ehsan et al. [25] proposed the Sparse Subspace Clustering (SSC) algorithm, which imposes an l_1 norm on self-expression matrix to make the feature in subspace as sparse as possible. But SSC is sensitive to the noise of data. To deal with this problem, René et al. [26] proposed the Low-Rank subspace clustering (LRR) method, which used kernel norm to optimize the self-expression matrix. However, some researchers [3,6,21,24,31] observed that traditional subspace clustering methods, both SSC and LRR, often exhibit limited performance when applied directly in the original subspace. Most classical methods can be formulated as:

$$\arg\min_{\theta \in R^{N \times N}} \|X - X\theta\|_F^2 + R(\theta), s.t.(\theta) = 0 \tag{1}$$

where $R(\theta)$ represent different norms. With the development of deep learning, more and more works leverage deep neural networks for data projection and clustering. They project the input data into a more suitable feature space by deep neural networks, and then applying clustering algorithms specifically designed for the projected representation [12]. In the past years, Deep subspace clustering network (DSC-NET) was proposed by Ji et al. [13], as shown in Fig. 1. DSC-NET inserted a fully connected linear as self-expression module between the encoder and decoder to generate the suitable subspace clustering coefficient. Inspired by low-rank representation, Kheirandishfard et al. [14] extended DSC-net and proposed Deep Low-Rank Subspace Clustering (DLRSC) to enable subspace clustering using the low-rank features that existed in original data separate samples. For a given data matrix $X \in R^{N \times D}$, the deep subspace clustering network can be described by the formulation:

$$l_\theta = \frac{1}{2}\left\|X - \bar{X}\right\|_F^2 + \varphi_1 \|\theta\|_p + \frac{\varphi_2}{2}\|Z - Z \cdot \theta\|_F^2, s.t. diag(\theta) = 0 \tag{2}$$

where Z is the latent features obtained after convolutional auto-encoder that its different classes of features belong to different subspaces.

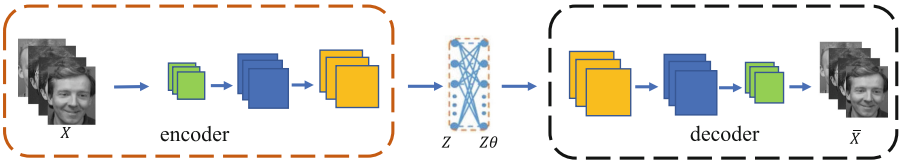


Fig. 1. The framework of Deep Subspace Clustering Network.

2.2 Self-supervised Deep Subspace Clustering

Self-supervised learning [12,15,17] can be broadly classified into two categories: generative and discriminative methods. Generative models take the original data

as input to project it into the latent space and then perform pixel-level reconstruction. Since depth subspace clustering relies solely on the reconstruction of a single pixel of the auto-encoder to map the data into the latent space. Therefore, many researchers leverage self-supervised learning methods to acquire additional auxiliary information and optimize features in the latent subspace. By designing pretext tasks that do not require human annotations, these approaches aim to exploit the inherent structure and patterns within the data to learn meaningful representations. Inspired by self-supervised learning, Zhou et al. [31] made further advancements to the Deep Subspace Clustering (DSC) model by introducing an adversarial approach known as Discriminative Adversarial Subspace Clustering (DASC). Then Yu et al. [29] proposed DSC-DAG, which leverages the dual generative adversarial networks (GANs) for learning latent features of the input data through an adversarial training process. In addition, Zhang et al. [30] proposed $S^2ConvSCN$ to optimize the self-expression model using the auxiliary information contained in the spectral clustering fused into the subspace clustering. Inspired by contrastive learning, Peng et al. [3] proposed deep contrastive subspace clustering (DSCSC) network, which enhanced the performance of subspace clustering. Chen et al. [20] proposed DSCNSS, which used the supervised information provided by clustering pseudo-label to improve network performance.

3 Method

In this section, we provide a comprehensive overview and all the necessary details of our model.

3.1 Architecture of DSSCT

Different from previous work, DSSCT consists of three modules: a) Triple Autoencoder Module (TAM), which consists of a triplet generation module and stacked convolutional auto-encoder. TAM can reconstruct triple data and capture the latent feature; b) Triplet Contrastive Module (TCM) projects the latent feature into a triple contrastive subspace to capture the relationship between positive pairs and negative pairs; c) Dual Self-Expression Fusion Module (DSFM), which incorporates dual self-expression layers to enhance the discriminant ability of the model. In the following, we will provide detailed explanations of each module incorporated in our model and show our model in Fig. 2.

Triple Auto-encoder Module (TAM). We first introduce the basic module TAM in DSSCT. TAM contains the triple data generation module and the triple auto-encoder module. In the triplet data generation module, we can easily generate the triplet data through data augmentation. Specifically, for given data $X_{anc} = [x_1...x_i...x_n] \in R^{N\times D}$, we use random data augmentation strategy $\tau^i(.)$ for each x_i to get augment data $\hat{x_i} = \tau^i(x_i)$. 2020 Mahdi et al. [16] shows that selecting an appropriate data augmentation strategy plays an

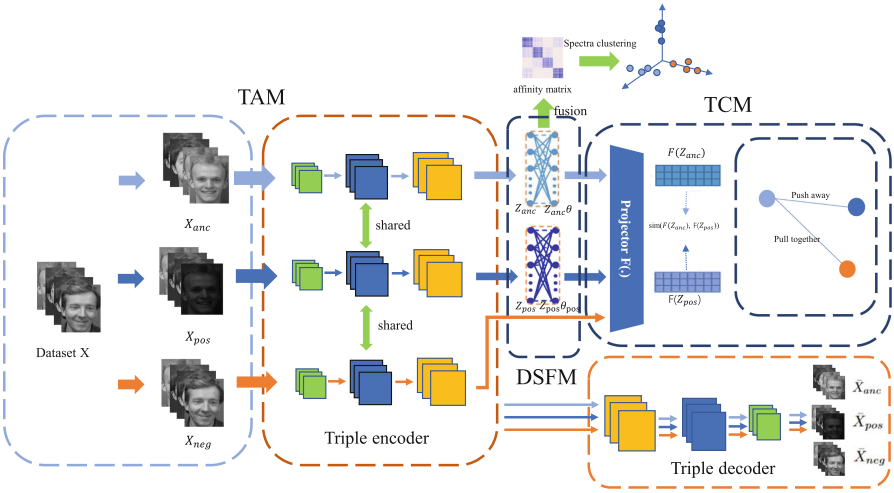


Fig. 2. The network architecture of the proposed DSSCT method, which consists of three modules divided by dotted line with different colors. The TAM is used to learn the latent embedded representation $Z_{anc}, Z_{pos}, Z_{neg}$, then DSFM combines Z_{anc}, Z_{pos} with θ, θ_{pos} to reconstruct the subspace latent representation. At the same time, TCM guides framework to learn the discriminative feature.

important role in improving subsequent task performance. Therefore, we follow the same augmentation strategy to construct our positive and negative pairs. In this work, six types of data augmentation methods are used i.e., Posterize, Sharpness, FlipLR, ShearX, TranslateY and Contrast. Then we can easily obtain positive sets $X_{pos} = \{\hat{x}_1 ... \hat{x}_i ... \hat{x}_n\}$ and negative sets $X_{neg} = \left\{\hat{x}_1^{'} ... \hat{x}_i^{'} ... \hat{x}_n^{'}\right\}$. Then triple data generation module combines the three sets to get triplet data $T = \{X_{anc}, X_{pos}, X_{neg}\}$. For a specific sample x_i, we use $\{x_i, \hat{x}_i\}$ as a positive pair and choose $\left\{x_i, \hat{x}_i, \hat{x}_i^{'}\right\}$ as a group of triplet data, where x_i is anchor sample, $\hat{x}_i$ is a positive sample and $\hat{x}_i^{'}$ is negative sample selected different from x_i and $\hat{x}_i$. Subsequently, the triple auto-encoder module, which consists of a few shared auto-encoders, maps the triplet data T into the latent subspace and gets the reconstructed data $\bar{X}_{anc}, \bar{X}_{pos}, \bar{X}_{neg}$ in the same triplet subspace. Finally, TAM network parameters are updated through the process of backpropagation with the following loss function:

$$l_{rec} = \frac{1}{2}\left(\left\|X_{anc} - \bar{X}_{anc}\right\|_F^2 + \left\|X_{pos} - \bar{X}_{pos}\right\|_F^2 + \left\|x_{neg} - \bar{X}_{neg}\right\|_F^2\right) \quad (3)$$

Triplet Contrastive Module (TCM). To preserve both local and overall structural information of the data, we propose a novel joint contrast loss function that combines the triplet loss and the temperature cross-entropy loss. This joint loss function encourages the model to learn an affinity matrix that captures more semantic information. Specifically, under the limitation of *margin*, triplet

loss enforces a higher similarity between samples X_{anc} and X_{pos} compared to the negative pair $\{X_{anc}, X_{neg}\}$ i.e., $s(X_{anc}, X_{pos}) - s(X_{anc}, X_{neg}) > margin$. Furthermore, the temperature cross-entropy loss can also be seen as soft triplet loss but with an adaptive margin, which will compensate for the limitation of triplet loss with a fixed margin. To alleviate the information loss caused by the triplet contrastive module, we introduce a projector $F(\cdot)$ to transform the triple latent features into another latent subspace, rather than directly perform triple contrastive on the original latent subspace. Note that $F(\cdot)$ has consisted of two-layer fully connected nonlinear MLP. The triplet contrastive loss function can be formulated by Eq. 6:

$$l_{triple} = max(F(Z_{anc}) \cdot F(Z_{pos}) - F(Z_{anc}) \cdot F(Z_{neg}) + margin, 0) \tag{4}$$

$$l_{Ce} = \log(\frac{exp(F(Z_{anc}) \cdot F(Z_{pos})/\iota)}{exp(F(Z_{anc}) \cdot F(Z_{pos})/\iota) + exp(F(Z_{anc}) \cdot F(Z_{neg})/\iota)}) \tag{5}$$

$$l_{tripleCe} = \xi_1 \cdot l_{triplet} - \xi_2 \cdot l_{Ce} \tag{6}$$

where ι is temperature parameter that control the softness, margin is similarity parameter that restraint the triplet data, ξ_1, ξ_2 are trade-off parameters.

In TCM, the reason that we combine the triplet loss and the temperature cross-entropy loss can be summarized as follows: a) Triplet loss could preserve local structure information and maintain a stable self-expression metric. b) Temperature cross-entropy loss can be regarded as soft triplet loss, which will compensating for the limitation of the fixed margin. c) The joint loss function will reduce the influence of negative pairs and enhancing the performance of DSSCT.

Dual Self-expression Fusion Module (DSFM). The dual self-expression fusion module is designed to fuse dual self-expression matrices and get discriminant features. With the constraint in Eq. 6, our model will follow the subspace-preserving property in the latent space to construct the self-expression layer. Different from other works, we believe that the positive sample pairs not only lie in the same subspace, but also can express each other in the latent subspace i.e., $Z_{anc} = Z_{anc} \cdot \theta, Z_{pos} = Z_{pos} \cdot \theta_{pos}$. Therefore, we use a weighted fusion strategy to obtain the shared self-expression matrix $C = \zeta_1 \cdot \theta \oplus \zeta_2 \cdot \theta_{pos}$, where ζ_1 and ζ_2 are hyper-parameters, $\oplus$ denotes an addition with different weights. The loss function of the dual self-expression fusion module can be described by the following equation:

$$\begin{aligned} l_{self} = \frac{1}{2}(\alpha \left\| Z_{anc} - Z_{anc} \cdot \theta \right\|_F^2 + (1-\alpha) \left\| Z_{pos} - Z_{pos} \cdot \theta_{pos} \right\|) + (\beta \left\| \theta \right\|_F \\ + (1-\beta) \left\| \theta_{pos} \right\|_F), s.t. diag(\theta) = 0, diag(\theta_{pos}) = 0 \end{aligned} \tag{7}$$

where α and β are different trade-off parameters that apply different attention to the dual self-expression matrix.

3.2 Optimization and Training Strategy

Our framework can be regarded as a 'multi-task' model, so it is still difficult to train the million-parameters network model directly. Therefore, we divide the whole training strategy into two parts i.e., pre-training and fine-tuning phases to obtain faster convergence speed and better generalization ability. At first, we pre-train the auto-encoder using the triplet data with Eq. 2. Then we fine-tune the whole network with overall objective loss Eq. 8 and obtain the affinity matrix W through $W = \frac{|C|+|C|^T}{2}$.

$$l_{toal} = \lambda_1 l_{rec} + \lambda_2 lself + \lambda_3 l_{tripleCe} \tag{8}$$

where $\lambda_i, i \in (1, 3)$ is trade-off parameter of DSSCT, W is regarded as the input of spectral clustering algorithm to get clustering result. The optimization strategy of our model is presented in Algorithm 1.

Algorithm 1. Training procedure of our DSSCT-Net

Input: Data X, trade-off parameters λ_1,λ_2,λ_3,λ_4,α, β margin, temperature ι, maximum iteration T_{max}, and Test iteration T_{test}.
Pre-training: Pre-train auto-encoder via (5);run TAM module to initialize X_{pos} and X_{neg} ; initialize θ and θ_{pos} with $1.0e-8$;
Fine-tuning:
  while $iter < T_{max}$ **do**
    update X_{pos} and X_{nbeg} by running TAM module;
    run the encoder to extract feature;
    update all the network parameters via minimizing l_{toal} with Adam solver;
    if $iter\%T_{test} = 0$ **then**
      run the DSFM module to get affinity matrix $C = \omega_1 \cdot \theta \oplus \omega_2 \cdot \theta_{pos}$.
      run the Spectral Clustering on C.
    end if
  end while
Output: label Y.

4 Experiments

In this section, we conducted a comprehensive evaluation of our proposed method on four public benchmark datasets. To assess its effectiveness, we compared our approach against state-of-art subspace clustering methods, and the details of experience will be shown in Sect. 4.3, 4.4 and 4.5.

4.1 Datasets

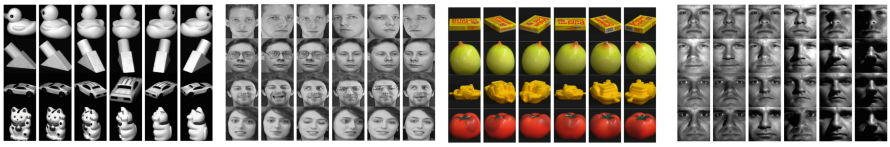
There are four public datasets included: COIL20 [19], ORL [22], COIL100 [19], Extended Yale B [10].

Coil20 dataset is a most common object dataset, which is consisted of 1440 grayscale images of 20 objects, each with the dimension of $32 \times 32 \times 1$.

ORL is a widely adopted face dataset in the field of feature extraction and image clustering. It comprises facial images collected from 40 distinct individuals. Each individual's face is represented by 10 images captured from different viewing angles, resulting in a total of 400 images on the dataset, we down-sample the dimension from 112×92 to 32×32.

Coil100 dataset comprises 1440 RGB images of 100 different objects. Following DSCN [13], we down-sample the dimension of those RGB images from 128×128 to 32×32 and use their grayscale version.

Extended Yale B is also a popular face dataset. It contains 38 individuals with 64 images, each recorded from different angles, for a total of 2432 face images. According to previous work, we down-sampled dimension to $48 \times 48 \times 1$. Some images of the dataset are shown in Fig. 3.



(a)Coil20 (b)ORL (c)Coil100 (d)Extended Yale B

Fig. 3. Sample images of the dataset used in the paper

4.2 Evaluation Indicators

In our experience, we employed two widely adopted performance metrics i.e., Accuracy (ACC) [23] and Normalized Mutual Information (NMI) [2], to assess the effectiveness of our method. ACC is used to evaluate the clustering performance, and it can be described by the following formulation:

$$\psi(a,b) = \begin{cases} 1 & whilea \neq b \\ 0 & whilea = b \end{cases} \tag{9}$$

$$ACC = 1 - argmin \frac{\sum_{i=1}^{N} \psi(y_i, \hat{y}_i))}{N}, s.t.(y_i, \hat{y}_i) \epsilon K \tag{10}$$

where N is the number of images, x_i denotes a sample that input the model, $\hat{y}_i$ s the prediction label output by the model, y_i is the ground truth of x_i, $\psi(.)$

Table 1. DSSCT-net clustering performance on Coil20 and Coil100 datasets

Methods	Coil20		Coil100	
	ACC	NMI	ACC	NMI
SSC	0.8631	0.8892	0.5510	0.5841
AE+SSC	0.8711	0.8747	0.4607	0.4871
LRR	0.8118	0.8603	0.4018	0.4721
ENSC	0.8760	0.8952	0.5732	0.5924
EDSC	0.8371	0.8828	0.6187	0.6751
SSC-OMP	0.6410	0.7412	-	-
DSC-NET-l1	0.9314	0.9353	0.6638	0.6720
DSC-NET-l2	0.9368	0.9408	0.6904	0.7015
LRSC	0.7416	0.8452	0.4933	0.5810
DLRSC	0.9708	-	0.7186	-
DSSCN	0.9799	-	0.7253	-
$S^2ConSCN$-l_1	0.9786	-	-	-
$S^2ConSCN$-l_2	0.9767	-	-	-
DSCSC	0.9788	0.9742	-	-
DSCNSS-l_1	0.9606	-	0.6966	-
DSCNSS-l_2	0.9624	-	0.7142	-
DSSCT (ours)	**0.9833**	**0.9824**	**0.7428**	**0.7528**

represent discriminant function, K is the number of subspaces. NMI is another commonly used in clustering. It accounts for the clustering quality in terms of both cluster purity and cluster completeness, and it can be formulated by Eq. 11:

$$NMI(Y,\hat{Y}) = 2N\frac{\sum_{i=1}^{Y}\sum_{j=1}^{\hat{Y}} m_{ij}\log(\frac{Nm_{ij}}{m_i m_j})}{\sum_{i=1}^{Y}\sum_{j=1}^{\hat{Y}} m_i m_j \log(\frac{m_i}{N})\log(\frac{m_j}{N})} \tag{11}$$

where $Y, \hat{Y}$ are the collections of y_i and $\hat{Y}_i$, m_i is the number of i-th class and m_j is the prediction number of j-th class. m_{ij} is the number of predicted categories that do not match the true labels. Note that these metrics provide robust and objective measures for evaluating our model, and both have a range of [0, 1], where the value more closer to 1 indicates better performance.

4.3 Coil20 and Coil100 Clustering

To evaluate the effectiveness of our model, we performed experiments on two object datasets and compared with several classic and recent methods. According to the previous work, we down-sample images to 32×32 and use the same experimental setup as DSCN. Specifically, the auto-encoder consists of one layer of convolutional layers with 15 and 50 channels, and the kernel sizes are 3×3 and 5×5 respectively, and the network setting is shown in Table 4. For Coil20 (K = 20) dataset, the parameter of our model used as follows: $\lambda_1 = 10.0$,

$\lambda_2 = 75.0$, $\lambda_3 = 20.0$, $\lambda_4 = 10.0$, $\iota = 0.5$, margin $=0.25$. The Coil100 dataset (K = 100) has a higher number of categories compared to COIL20, so we use the following trade-off parameters: $\lambda_1 = 10.0$,$\lambda_2 = 75.0$, $\lambda_3 = 20.0$, $\lambda_4 = 10.0$, $\iota = 0.5$, margin $= 0.2$. Table 1 summarizes the clustering results on two object datasets. It is able to see that the accuracy of our method is higher than DSC-net-l_2 by 4.58% and 4.18% on the two datasets, and both better than the current self-supervised methods. This demonstrates the excellence of our model on the item dataset.

Table 2. DSSCT-net clustering performance on ORL and Extended Yale B datasets

Methods	ORL		EYale B	
	ACC	NMI	ACC	NMI
SSC	0.7425	0.8459	0.7354	0.7796
AE+SSC	0.7450	0.8824	0.7475	0.7764
LRR	0.8110	0.8603	0.8499	0.8636
ENSC	0.7525	0.8540	0.7537	0.7915
EDSC	0.7038	0.7799	0.8814	0.8835
SSC-OMP	0.7100	0.7952	0.7372	0.7803
DSC-NET-l_1	0.8550	0.9032	0.9681	0.9687
DSC-NET-l_2	0.8600	0.9034	0.9733	0.9703
LRSC	0.7200	0.8156	0.7913	0.8264
DLRSC	-	-	0.9753	-
DSSCN	0.8850	-	0.9432	-
$S^2ConSCN$-l_1	0.8875	0.9214	0.9848	-
$S^2ConSCN$-l_2	0.8950	0.9238	0.9844	-
DSCSC	0.9075	0.9444	0.9864	0.9815
DSCNSS-l_1	0.8868	-	0.9792	-
DSCNSS-l_2	0.8921	-	0.9815	-
DSSCT(ours)	**0.9084**	**0.9519**	**0.9823**	**0.9756**

We also visualized the loss functions during alternation training. Figure 4 shows that the loss of each model gradually decrease until it become stable. With the optimization of other sub-loss, the affinity Loss demonstrates that affinity matrix θ strives to learn information as much as possible in the subspace.

4.4 ORL and Extend Yale B Clustering

Then we evaluate our method on the face image ORL and Extend Yale B. In the process of experience, the structure of the auto-encoder still consistent

Table 3. Accuracy clustering result (%) on Coil20 dataset with different combinations of modules

	Coil20		Coil100	
	ACC	NMI	ACC	NMI
TAM	0.9444	0.9448	0.6938	0.7086
TAM with TCM	0.9826	0.9819	0.7322	0.7455
TAM and TCM with DSFM	**0.9833**	**0.9824**	**0.7428**	**0.7528**

with previous work. Both encoder and decoder have three-layer convolutional structure with (5, 3, 3) and (10, 20, 30) channels, and the kernel sizes are 5×5 and 3 respectively. As shown in Table 2, our model achieves an accuracy of 90.84%, which is better than the DSC-net-l_2 method by 4.84% and higher than the compared methods. On the Extend Yale B dataset, we achieve an accuracy of 98.23%, which also has perfect performance. It is worth noting that we do not specifically train the pre-trained model, so there still have some potential to improve the results on the face dataset. Table 4 shows the details of the network on different real-word datasets.

Table 4. The network setting for different dataset

	encoder-1	encoder-2	encoder-3	self-expression	decoder-1	decoder-2	decoder-3
Coil20	3×3	-	-	1440×1440	-	-	3×3
Coil100	5×5	-	-	7200×7200	-	-	5×5
ORL	5×5	3×3	3×3	400×400	3×3	3×3	5×5
EYale B	5×5	3×3	3×3	2432×2432	3×3	3×3	5×5

4.5 Ablation Experiments

In order to assess the positive impact of different modules on DSSCT (Deep Self-Supervised Subspace Clustering with Triple Data), some ablation experiments are conducted. We performed ablation experiments on two object datasets and the results of the experiment are summarized in Table 3. In addition, we have visualized the block-diagonal structure matrix generated by our method and compared them with DSC-Net on Coil20, Coil100 and ORL datasets. Specifically, the affinity matrix is a mutual-expressed matrix that has high-dimensional data in a particular low-dimensional subspace. Therefore, the affinity matrix with subspace-preserving properties should have a distinct block diagonal structure and the more distinct block diagonal the better discriminant. As shown in Fig. 6, our model has obtained a more discriminative block diagonal matrix on different datasets.

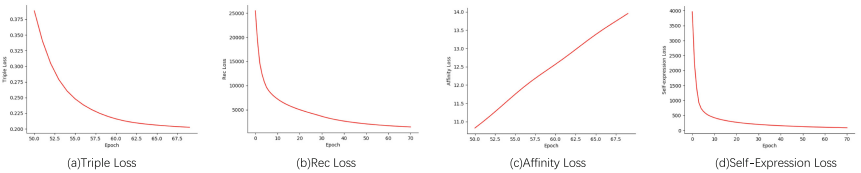


Fig. 4. The optimized trend of sub-loss (a) Triple Loss; (b) Rec Loss; (c) Affinity Loss (d) Self−Expression Loss

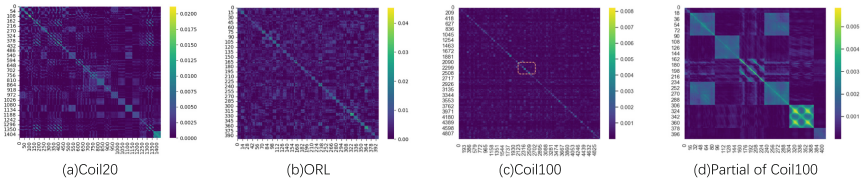


Fig. 5. The block-diagonal structure matrix of DSCN on different datasets

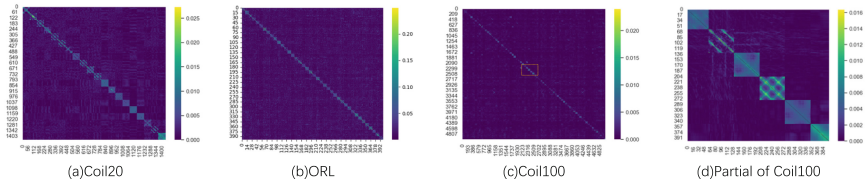


Fig. 6. The block-diagonal structure matrix of our method on different datasets

5 Conclusion

In this paper, we proposed a novel framework called DSSCT-net, which integrated the triplet contrastive loss into our model. We introduce a dual self-expression layer structure to enhance the discriminant of our model. The experimental result shows that our model gets excellent performance on many real-word datasets and is competitive compared with other state-of-art methods. In the future, we will continue improve the robustness and efficiency of our model.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 62101213, No. 62103165, 62302191, the Natural Science Foundation of Shandong Province, China, under Grant No. ZR2020QF107, No. ZR2020MF137, ZR2023QF001, Development Program Project of Youth Innovation Team of Institutions of Higher Learning in Shandong Province

References

1. Bradley, P.S., Mangasarian, O.L.: K-plane clustering. J. Global Optim. **16**, 23–32 (2000)
2. Cai, D., He, X., Han, J.: Locally consistent concept factorization for document clustering. IEEE Trans. Knowl. Data Eng. **23**(6), 902–913 (2010)
3. Chen, C., Lu, H., Wei, H., Geng, X.: Deep subspace image clustering network with self-expression and self-supervision. Appl. Intell. **53**(4), 4859–4873 (2023)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
5. Chen, X., Zhexue Haung, J., Nie, F., Chen, R., Wu, Q.: A self-balanced min-cut algorithm for image clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2061–2069 (2017)
6. Chen, Y., Xiao, X., Zhou, Y.: Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix. Pattern Recogn. **106**, 107441 (2020)
7. Dang, Z., Deng, C., Yang, X., Huang, H.: Multi-scale fusion subspace clustering using similarity constraint. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6658–6667 (2020)
8. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2765–2781 (2013)
9. Gear, C.W.: Multibody grouping from motion images. Int. J. Comput. Vis. **29**, 133–150 (1998)
10. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
11. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
12. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
13. Ji, P., Zhang, T., Li, H., Salzmann, M., Reid, I.: Deep subspace clustering networks. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
14. Kheirandishfard, M., Zohrizadeh, F., Kamangar, F.: Deep low-rank subspace clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 864–865 (2020)
15. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: a framework and review. IEEE Access **8**, 193907–193934 (2020)
16. Li, C., Yang, C., Liu, B., Yuan, Y., Wang, G.: LRSC: learning representations for subspace clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 8340–8348 (2021)
17. Liu, Y., et al.: Graph self-supervised learning: a survey. IEEE Trans. Knowl. Data Eng. **35**(6), 5879–5900 (2022)
18. Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 347–360. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33786-4_26
19. Nene, S.A., Nayar, S.K., Murase, H., et al.: Columbia object image library (COIL-20) (1996)

20. Peng, B., Zhu, W.: Deep structural contrastive subspace clustering. In: Asian Conference on Machine Learning, pp. 1145–1160. PMLR (2021)
21. Peng, X., Zhu, H., Feng, J., Shen, C., Zhang, H., Zhou, J.T.: Deep clustering with sample-assignment invariance prior. IEEE Trans. Neural Net. Learn. Syst. **31**(11), 4857–4868 (2019)
22. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, pp. 138–142. IEEE (1994)
23. Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y.: SpectralNe: spectral clustering using deep neural networks. arXiv preprint arXiv:1801.01587 (2018)
24. Valanarasu, J.M.J., Patel, V.M.: Overcomplete deep subspace clustering networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 746–755 (2021)
25. Vidal, E.E.R.: Sparse subspace clustering. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 6, pp. 2790–2797 (2009)
26. Vidal, R., Favaro, P.: Low rank subspace clustering (LRSC). Pattern Recogn. Lett. **43**, 47–61 (2014)
27. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)
28. Yang, A.Y., Rao, S.R., Ma, Y.: Robust statistical estimation and segmentation of multiple subspaces. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, pp. 99–99. IEEE (2006)
29. Yu, Z., Zhang, Z., Cao, W., Liu, C., Chen, C.P., Wong, H.S.: Gan-based enhanced deep subspace clustering networks. IEEE Trans. Knowl. Data Eng. **34**(7), 3267–3281 (2020)
30. Zhang, J., et al.: Self-supervised convolutional subspace clustering network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5473–5482 (2019)
31. Zhou, P., Hou, Y., Feng, J.: Deep adversarial subspace clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1596–1604 (2018)