Image Clustering Based on Multi-Scale Deep Maximize Mutual Information and Self-Training Algorithm

SIQUAN YU\textsuperscript{1,2,3}, GUIPING SHEN\textsuperscript{4}, PEIYAO WANG\textsuperscript{5}, YUNING WANG\textsuperscript{6}, AND CHENGDONG WU\textsuperscript{7}, (Member, IEEE)

\textsuperscript{1}College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
\textsuperscript{2}State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China
\textsuperscript{3}Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
\textsuperscript{4}College of Artificial Intelligence, Yangtze University, Fuzhou 350108, China
\textsuperscript{5}Shenyang Institute of Technology, Shenyang 110000, China
\textsuperscript{6}PLA 32681, Shenyang 110000, China
\textsuperscript{7}Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China

Corresponding author: Guiping Shen (guipingshen@yeah.net)

ABSTRACT Image clustering is a complex procedure that is significantly affected by the choice of the image representation. Generally speaking, image representations are generated by using handcraft features or trained neural networks. When dealing with high dimension data, these two representation methods cause two problems: i) the representation ability of the manually designed features is limited; ii) the non-representative and meaningless feature of a trained deep network may hurt the clustering performance. To overcome these problems, we propose a new clustering method which efficiently builds an image representation and precisely discovers the cluster assignments. Our main tools are an unsupervised representation learning method based on Deep Mutual Information Maximization (DMIM) system, and a clustering method based on self-training algorithm. Specifically speaking, to extract the informative representation of image data, we derive the maximum mutual information theory and propose a system to learn the maximum mutual information between the input images and the latent representations. To discover the clusters and assign each image a clustering label, a self-training mechanism is applied to cluster the learned representations. The superiority and validity of our algorithm are verified in a series of real-world image clustering experiments.

INDEX TERMS Representation learning, image clustering, mutual information maximization, self-training algorithm.

I. INTRODUCTION Clustering, a vital research topic in the field of data science and unsupervised learning, which aim to classify elements into categories on the basis of their similarity [1]. The clustering problem has been extensively studied in the past decades. However, the performance of standard clustering algorithms is adversely affected when dealing with high-dimensional data [2]. Because image is a kind of high-dimensional data, image clustering is always a challenging task in computer vision and machine learning [3].

Generally speaking, the traditional image clustering methods such as k-means++ [4], gaussian mixture model [5] and spectral clustering [6] group images on handcrafted features and treat feature extraction and clustering separately. Based on this insight, many attempts have dedicated to developing suitable clustering feature extracting techniques such as manually designed feature descriptors, including Bag of Feature (BOF) [7], Histogram of Oriented Gradient (HOG) [8], Principal Component Analysis (PCA) [9] and Scale-Invariant Feature Transform (SIFT) [10]. However, the representation ability of the manually designed feature methods is limited; the traditional clustering methods may be invalid due to the influence of some messy variable. They mostly suffer from appearance variations of scenes and objects when...
dealing with image data. How to automatically learn an image representation to capture the image information is a crucial problem that needs to be solved in image clustering tasks.

In recent years, some novel representation learning methods have emerged, such as manifold alignment [11], dictionary learning [12] and deep neural network [13]. Among them, deep neural networks have been successfully applied to various supervised learning tasks [14]. The reasons for the success of deep neural networks is that it can automatically learn the essential representation of images by constructing networks with multiple hidden layers, and train the network with a large number of data [15]. Motivated by this, some studies are devoted to cluster images based on deep neural networks, that is deep clustering [16, 17].

Deep clustering learns deep representations that favor clustering tasks using neural networks. In most of previous deep clustering studies, they usually train a deep generation model. For example, Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Generative Adversarial Network (GAN), to reduce the dimension of image data. Then, part of the trained generation model is fine-tuned by using the clustering algorithms, and the generation model provides deep feature to the clustering algorithms to discover clusters [18, 19]. This two-stage training scheme has been successfully applied to many clustering works [2], [20]–[22]. However, the clustering results may suffer from the invalid image representation. Generation models use reconstruction loss to learn image representation which is not an optimal choice for clustering tasks.

Most recently, some deep clustering methods try to combine image representation with clustering learning [23], [24]. Generally, they construct neural networks and use clustering algorithm as loss function to train network directly. From the perspective of representation learning, one-stage clustering scheme is more reasonable while it learns image representation and clustering information simultaneously. However, one-stage clustering also involves a non-representation problem, that is, the clustering algorithm does not match the deep learning. This may lead to the learned image representation only focuses on clustering, and lacks the essential information of the image, and causes the phenomenon of degenerate solution.

To overcome the aforementioned problems, this paper focuses on the establishment of an image clustering method based on the Deep Mutual Information Maximization (DMIM) and a self-training algorithm. Specifically speaking, we first derive the Mutual Information Maximization (MIM) theory into deep neural network, and propose a Deep Mutual Information Maximization (DMIM) system to learn an informative representation of image data. To discover the clusters of the input images, we assign each image a clustering label, and adopt a self-training algorithm to fine-tune the DMIM system to obtain a more friendly clustering representation. We conduct series of experiments to verify the effectiveness of our algorithm, and the performance of the proposed algorithm outperforms the newest opponent in a large margin.

The main contributions of this paper can be concluded in three aspects. Firstly, we propose a novel maximum mutual information system based on statistical learning theory, and use it to learn an informative image representation. Secondly, we incorporate the learned image representation to a self-training algorithm to realize image clustering. Thirdly, we conduct extensive experiments on four real-world datasets to verify the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. In Section II, we introduce the related work of our paper. Section III proposes the clustering algorithm as well as some details of the algorithm. Section IV provides a series of experiments to analyze the effect of parameters and verify the superiority of the proposed algorithm. We conclude this paper in the last section.

II. RELATED WORK

A. DEEP CLUSTERING

Deep clustering refers to clustering with the related algorithm of deep learning, which has been widely concerned and studied in recent years. The existing deep clustering algorithms are mainly divided into two categories: (I) a two-stage work that applying clustering after a representation is learned. (II) a one-stage work that jointly optimization the representation learning and clustering.

Some two-stage methods usually train a generation model at the first stage. Then, the trained generation model acts as a feature extractor and uses clustering algorithm to obtain the clustering results. For instance, Guo et al. propose Convolutional Auto-Encoder (CAE), using k-means algorithm to cluster the auto-encoder’s image representation [20]. Ghasedi Dizaji et al. propose Stacked Auto-Encoder (SAE) algorithm which first train an AE, and uses relative entropy as a loss function training encoder to obtain clustering results [2]. Xie et al. propose Deep Embedded Clustering (DEC) which starts with a pre-training phase using only the reconstruction loss and then improves the clustering ability of the representation by optimizing in a self-supervised manner [25]. Peng et al. propose a novel clustering method by minimizing the discrepancy between pairwise sample assignments for each data point [26]. Gaussian Mixture Variational Autoencoders (GMVAE) is a representative generation-based clustering algorithm that incorporates gaussian distribution to variational AutoEncoder [22]. Categorical Generative Adversarial Networks (CatGAN) is another clustering algorithm that based on generative models. It is based on an objective function that trades-off mutual information between observed examples and their predicted categorical class distribution, against robustness of the classifier to an adversarial generative model [19]. The disadvantage of two-stage methods is the unsuitable problem between image representation and clustering. Since the target of generation model is to make the generated image close to the input image in visualization, while clustering aims to reduce all possible variations into several templates [27], [28]. This difference makes generation
model unsuitable to participate in deep clustering directly as the important discriminative information will be lost.

One-stage methods combine the image representation with the clustering learning process. Joint Unsupervised Learning of Deep Representations and Image Clustering (JULE), Structured AutoEncoders for Subspace Clustering (SASC) and Deep Adaptive Clustering (DAC) are three representative image clustering methods that simultaneously learn the image representation and the clustering results. JULE proposes a recurrent framework for joint unsupervised learning of deep representations and image clusters [23]. DAC defines an effective objective and proposes an adaptive mechanism to realize image clustering [24]. The defined objective function is used to update the parameters of a convolutional network by selecting highly confidence image pairs and the cluster assignment is integrated into classification labels. SASC proposes a clustering method based on the subspace clustering theory and a local preserving scheme. It improves the traditional subspace clustering methods by using an autoencoder to guarantee that the learned representations preserve the local and global subspace structure [29]. The effectiveness of these learning schemes has been proved in theory and practical experiments. However, there are two crucial factors that affect the stability and effectiveness of these two algorithms. On one hand, the initialization of convolutional network is an important factor that affects the performance of DAC and JULE. On the other hand, with the training going on, the local structure preservation of representation cannot be guaranteed. The image representation in the distorted feature space may not be suitable for clustering.

B. MUTUAL INFORMATION MAXIMIZATION

Since the definition of mutual information is based on the information entropy, we first briefly introduce the corresponding concepts of information entropy. Information entropy is the average rate at which information is produced by a stochastic source of data [30]. The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value. Given \(X = \{x_1, x_2, \ldots, x_n\}\), the information entropy is defined as,

\[
H(X) = - \int p(x) \log p(x) dx,
\]

(1)

where \(p(x)\) is the probability density of \(X\).

If \(Z\) is the latent variable of \(X\), mutual information (MI) is a measure of the reduction of uncertainty in \(X\) due to the knowledge of \(Z\) [31]. In information theory, given two random variables \(X\) and \(Z\) with the joint distribution \(p(x, z)\) and the marginal distribution \(p(x)\) and \(p(z)\), MI between \(X\) and \(Z\) can be calculated as follows,

\[
I(X; Z) = H(X) - H(X|Z)
= - \int \int p(x, z) \log \frac{p(x,z)}{p(x)} dx dz + \int \int p(x, z) \log p(x|z) dx dz.
\]

(2)

Therefore, MI can be defined as follows,

\[
I(X; Z) = D_{KL}(p(x,z)||p(x)p(z)),
\]

(3)

where \(D_{KL}(\cdot)\) denote Kullback-Leibler divergence (KL divergence for short).

Mutual Information Maximization (MIM) is based on the definition of IE and MI, and is a technique for maximizing the average mutual information between two variables [32], [33]. In this paper, we maximize the mutual information between image \(X \in R^D\) and representation \(Z \in R^d\), and \(d\). Thus, we realize the dimensionality reduction of \(X\), and extract the informative representation of image data.

III. METHOD

In this section, we first derive the mutual information maximization to deep neural network and establish a deep mutual information maximization system. Then, a self-training method is involved in the training of image representation to obtain a representation suitable for clustering. Finally, we provide the network architecture and the detailed training procedure of the proposed algorithm.

A. DEEP MUTUAL INFORMATION MAXIMIZATION

As mentioned above, the mutual information between inputs \(X\) and their representations \(Z\) is defined by the KL-divergence (3), which can be decomposed as follows,

\[
I(X; Z) = D_{KL}(p(x,z)||p(x)p(z))
= \int \int p(z, x) \log \frac{p(x,z)}{p(x)p(z)} dx dz
= \int \int p(z|x)p(x) \log \frac{p(z|x)}{p(z)} dx dz,
\]

(4)

where \(p(x)\) denote the probability density of input data \(X\), \(p(z)\) is the probability density of representation \(Z\). We will often write densities like \(p(Z = z)\) as \(p(z)\) to save space. To learn the maximize mutual information between the input image and the image representation, we model \(p(z|x)\) as neural network, and assume that \(p(z)\) follows the standard normal distribution. This assumption is similar to what is done in VAE. Therefore, the objective function used to train the network \(p(z|x)\) can be defined as follows,

\[
\max_{p(z|x)} I(X; Z), p(z) \sim N(0, I).
\]

(5)

In [33], Deep InfoMax optimizes (5) by using an adversarial scheme. However, we observe that this scheme often leads to an instability problem in the in the process of network training. Different from Deep InfoMax, we directly add a restriction on the distribution of representation \(Z\), and the restriction is defined as follows,

\[
D_{KL}(p(z)||q(z)) = \int p(z) \log \frac{p(z)}{q(z)} dz.
\]

(6)
where \( q(z) \) is a prior distribution, which follows the standard normal distribution. Combining (5) with (6), the whole objective function is defined as follows,

\[
\min_{p(z|x)} \left\{ -I(X; Z) + \gamma D_{KL}(p(z)\| q(z)) \right\}, \tag{7}
\]

where \( \gamma \) is the balance coefficient.

To calculate the minimal optimization problem, we decompose (7) as follows,

\[
-I(X; Z) + \gamma D_{KL}(p(z)\| q(z)) \\
= -\int p(z|x)p(x) \log \frac{p(z|x)}{p(z)} dx dz + \gamma \int p(z) \log \frac{p(z)}{q(z)} dz \\
= -\int p(z|x)p(x) \left[ \log \frac{p(z|x)}{p(z)} + \gamma \log \frac{p(z)}{q(z)} \right] dx dz \\
= \int p(z|x)p(x) \left[ -(1+\gamma) \log \frac{p(z|x)}{p(z)} + \gamma \log \frac{p(z|x)}{q(z)} \right] dx dz \\
= -(1+\gamma) D_{KL}(p(z|x)p(x)\| p(z)) + \gamma \mathbb{E}_{x \sim p(x)} \left[ D_{KL}(p(z|x)\| q(z)) \right]. \tag{8}
\]

Therefore, (7) is equivalent to the following optimization problem,

\[
\min_{p(z|x)} \left\{ -\beta D_{KL}(p(z|x)p(x)\| p(z)p(x)) + \alpha \mathbb{E}_{x \sim p(x)} \left[ D_{KL}(p(z|x)\| q(z)) \right] \right\}. \tag{9}
\]

Because of KL-divergence is theoretically unbounded, we use Jensen-Shannon divergence (JS divergence for short) to instead of it. The objective function can be transformed as,

\[
\min_{p(z|x)} \left\{ -\beta D_{JS}(p(z|x)p(x), p(z)p(x)) + \alpha \mathbb{E}_{x \sim p(x)} \left[ D_{KL}(p(z|x)\| q(z)) \right] \right\}. \tag{10}
\]

In [34], f-GAN have proved that the minimization JS-divergence can be calculated as follows,

\[
\min_{p(z|x)} D_{JS}(p(z|x)p(x), p(z)p(x)) \\
= \max_{D(x,z)} \left\{ \mathbb{E}_{x \sim p(x)}[\log \sigma(D(x, z))] + \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(D(x, z)))] \right\}. \tag{11}
\]

where \( D(x, z) \) is a learnable parameter. Therefore, the objection function (10) can be rewritten as,

\[
\min_{p(z|x), D(x,z)} \left\{ \mathbb{E}_{x \sim p(x)}[\log \sigma(D(x, z))] + \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(D(x, z)))] + \alpha \mathbb{E}_{x \sim p(x)} \left[ D_{KL}(p(z|x)\| q(z)) \right] \right\}. \tag{12}
\]

To involve more comprehensive information of images, we combine the local and global mutual information loss, and define the following objective function,

\[
\min_{p(z|x), D_1(x,z), D_2(z,g)} \alpha L_p - \beta L_g - \gamma L_l, \tag{13}
\]

where \( D_1(x, z_g) \) and \( D_2(z, z_g) \) are the learnable parameters. \( L_g, L_l \) and \( L_p \) are the local mutual information loss function, global mutual information loss function and prior distribution of latent variable, respectively. The definitions of \( L_g, L_l \) and \( L_p \) are as follows,

\[
L_g = \mathbb{E}_{x \sim p(x)} \left[ \log \sigma(D_1(x, z_g)) \right] \\
+ \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(D_1(x, z_g)))].
\]
\[ L_1 = \mathbb{E}_{(x,z) \sim p(z|x)p(x)} \left[ \log \sigma(D_2(z; \theta)) \right] \]
\[ + \mathbb{E}_{(x,z) \sim p(z|x)p(x)} \left[ \log (1 - \sigma(D_2(z; \theta))) \right]. \]
\[ L_p = \mathbb{E}_{x \sim p(x)} \left[ D_{KL} \left( p(z|x) \mid \mid q(z) \right) \right]. \]

(14)

**B. NETWORK ARCHITECTURE**

As a practical applications of the ideas described above, we will now develop a system for maximizing the mutual information between the input \( X \) and the latent variable \( Z \) to obtain an image representation. For this purpose, we model \( p(z|x) \), \( D_1(x, z_q) \) and \( D_2(z_l, z_g) \) by three neural networks, that is, \( f_0(x) \), \( g_{\theta_1}(x, z_g) \) and \( g_{\phi_2}(z_l, z_g) \), where \( \theta, \phi_1 \) and \( \phi_2 \) denote the weights and biases parameters of the networks. The system we established is shown in Fig. 1. Specifically, \( f_0(x) \) maps the input \( x \) to the global latent variable \( z_g \), which can be implemented by several convolutional layers. For \( g_{\phi_1}(x, z_g) \), we first encode \( x \) into a vector, and concatenate this vector with the global representation \( z_g \). Then, we feed the connection vectors to several fully connection layers to realize \( g_{\phi_1}(x, z_g) \). The implementation of \( g_{\phi_2}(z_l, z_g) \) is similar to \( g_{\phi_1}(x, z_g) \) expect the connections are the global representation \( z_g \) and the local representation \( z_l \). The goal of the proposed system is to learn an encoder which maps input images to the informative representations. Next, we introduce the details of the complete loss function.

**C. COMPLETE LOSS FUNCTION**

All the aforementioned objects including global mutual information, local mutual information and prior matching loss function are jointed together. The complete loss function for learning image representation is defined as follows,

\[ \min_{\theta, \phi_1, \phi_2} -\beta \mathcal{L}_g - \gamma \mathcal{L}_l + \alpha \mathcal{L}_p \]  
(15) \]

and

\[ \mathcal{L}_g = \mathbb{E}_{x \sim p_{pos}} \left[ \log g_{\phi_1}(x, z_g) \right] \]
\[ + \mathbb{E}_{x \sim p_{neg}} \left[ \log (1 - g_{\phi_1}(x, \hat{z}_g)) \right]. \]
\[ \mathcal{L}_l = \mathbb{E}_{x \sim p_{pos}} \left[ \log g_{\phi_2}(z_l, z_g) \right] \]
\[ + \mathbb{E}_{x \sim p_{neg}} \left[ \log (1 - g_{\phi_2}(\hat{z}_l, z_g)) \right]. \]
\[ \mathcal{L}_p = \mathbb{E}_{x \sim p_{pos}} \left[ D_{KL} \left( p(z|x) \mid \mid q(z) \right) \right]. \]

(16)

where \( p_{pos} \) and \( p_{neg} \) denote the distributions of positive samples and negative samples. \( z_l \) and \( z_g \) are the local and global latent representations of positive samples. \( \hat{z}_l \) and \( \hat{z}_g \) are the local and global latent representations of negative samples, respectively.

The complete loss function consists of three components. The first and second terms in (15) denote the local and global mutual information loss function, which measure the information relevance between input image and representation. The third item in (15) denotes the prior loss which measure the errors between model prediction and the target variable.

Next, we introduce the implementation details of the proposed deep mutual information network. For the implementations of \( \mathcal{L}_g \) and \( \mathcal{L}_l \), the two expectations can be approximated by Monte Carlo sampling. Since the expectations of \( p(z|x)p(x) \) and \( p(z)p(x) \) can be realized by sampling on positive and negative samples, respectively [35]. To achieve this, we borrow the well known word2vec algorithm and adopt negative sampling trick to obtain positive and negative samples [36]. Specifically, we first randomly select a batch of samples from the dataset as positive samples. The positive samples follow the distribution \( p_{pos} \). Then, we shuffle the samples and take the disordered samples as negative samples. The negative samples follow the distribution \( p_{neg} \).

Similar to VAE, \( p(z|x) \) follows the normal distribution of mean \( u_k(x) \) and variance \( \sigma_k^2(x) \), \( p(z|x) \sim [u_k(x), \sigma_k^2(x)], k = 1, 2, \ldots, d \), that is,

\[ p(z|x) = \frac{1}{\prod_{k=1}^{d} \sqrt{2\pi \sigma_k^2(x)}} \exp \left( -\frac{1}{2} \frac{(z - u_k(x))}{\sigma_k(x)} \right). \]

(17)

Since \( q(z) \) follows the standard normal distribution \( q(z) \sim N(0, I) \). Thus, \( D_{KL} \left( p(z|x) \mid \mid q(z) \right) \) can be calculated as follows,

\[ D_{KL} \left( p(z|x) \mid \mid q(z) \right) = \frac{1}{2} \sum_{k=1}^{d} \left( \mu_k^2(x) + \sigma_k^2(x) - \ln \sigma_k^2(x) - 1 \right). \]

(18)

For implementation of \( \mathcal{L}_p \), \( \mu_k(x) \) and \( \sigma_k(x) \) are the outputs of network \( f_0(x) \). Note that the expectations of \( p_{pos} \) denotes that the prior loss only calculate in positive samples.

**D. CLUSTERING LOSS FUNCTION**

By maximizing the mutual information between the input image and the representation, the image representation with the most image information can be obtained. In this section, to mine the clustering characteristics of image representation, a self-learning module is proposed and integrated into the training of deep mutual information network.

The concept of self-training is derived from semi supervised learning. It first trains a classifier by using the known labeled samples, and then uses the trained classifier to evaluate the label of unlabeled samples [37]. For the unsupervised problem, the self-training can be transformed into two steps: calculating the initial label of samples and training the network with high confidence to correct the low confidence label. Therefore, it is very important to generate the initial labels as close to the real label distribution as possible. Inspired by the DEC algorithm, in this paper, we use self-training module as a tool to discover the clusters of the learned image representation. Specifically, we first use the well known student’s t-distribution and the learned maximization mutual information representation to calculate the pseudo label of each image, and get the pseudo label distribution of the image. Then, we establish a target distribution and use it to...
fine-tune the learned maximization mutual information network. The reasons why we incorporate the maximum mutual information representation to self-training algorithm contain two points: i) the initial pseudo labels of the self training algorithm should be as close as possible to the semantic labels of images, which means that the learned image representation should contain as much image information as possible; ii) the target distribution is established by enhancing the high confidence pseudo labels and weakening the low confidence pseudo labels.

In [25], DEC proposes a student’s t-distribution as a kernel to measure the similarity between the image representation \( z_i \) and the cluster centroid \( u_j \) as follows,

\[
l_{ij} = \frac{(1 + ||z_i - u_j||^2/v)^{-\frac{v+1}{2}}}{\sum_j (1 + ||z_i - u_j||^2/v)^{-\frac{v+1}{2}}},
\]

where \( z_i \) denotes the \( i \)-the image representation, \( u_j \) is initialized by K-means on representations learned by pre-train autoencoder, and \( v \) denote the defrees of freedom of the Student’s t-distribution. In particular, \( l_{ij} \) can be regarded as a pseudo label of each input image, and denote the probability of assigning sample \( i \) to cluster \( j \).

When we obtain the clustering assignment distribution \( Q \), Akin [25], we also propose a target distribution to optimize the image representation by learning from the high confidence pseudo labels. Specifically, the target distribution \( P \) is defined as follows,

\[
t_{ij} = \frac{t_{ij}^2/f_j}{\sum_{j'} t_{ij}'/f_j},
\]

where \( f_j = \sum_i l_{ij} \) denote the pseudo label frequencies. The goal of image clustering is to align Student’s t-distribution with the target distribution (20). Therefore, we define the following clustering loss function,

\[
L_c = D_{KL}(T||L) = \sum_{i} \sum_{j} t_{ij} \log \frac{t_{ij}^2}{l_{ij}}.
\]

By minimizing the distance between \( T \) and \( L \) distributions, the target distribution can guide the DMIM system to learn an image representation that is more suitable for image clustering.

### E. TRAINING AND CLUSTERING

Given a dataset \( X = \{x_1, x_2, \ldots, x_n\} \) with \( n \) samples that need to be clustered. The number of clustering \( K \) is a priori knowledge. Let the value of \( Z = \{z_1, z_2, \ldots, z_n\} \) is the image representations. We first sample a batch of images from the dataset \( X \) to construct positive sample set, and shuffle the image to construct negative sample set. Then, we train the proposed DMIM system by optimizing the object proposed in (15) to obtain the image representation \( Z \). Finally, we calculate the clustering pseudo labels by using (19), and update the DMIM system by using the self-training objective function (21). The entire training procedure of the proposed algorithm is presented in Algorithm 1.

---

**Algorithm 1 Clustering Based on DMIM and Self-Training**

**Input:** Input images \( X := \{(x_i)_{i=1}^m\}; \) The number of clustering \( K \); The batch size \( m = 128 \) and the learning rates \( \rho_1 = 0.005 \) and \( \rho_2 = 0.0001 \); The parameters \( \alpha = 0.01 \), \( \beta = 0.5 \), \( \gamma = 0.5 \) and \( \nu = 1 \).

**Output:** The clustering labels \( C \).

1. **Construction the positive sample set and the negative sample set:**
   2. Randomly select a batch of images as positive samples, and shuffle them as negative samples.
2. **Train the DMIM model to obtain image representation:**
   3. for \( i_1 \) in epochs1 do
      4. Compute the image representation \( Z \).
      5. Update \( \theta, \phi_1 \) and \( \phi_2 \) by minimizing (15) with the learning rate \( \rho_1 \).
   7. end for
3. **Self-training for the optimization of the network:**
   8. Cluster the image representations by using k-means++ and initialize clustering centroids \( u_j, j = 1, 2, \ldots, K \).
   9. Calculate \( T \) and \( L \) by using (19) and (20).
11. for \( i_2 \) in epochs2 do
    12. Update \( \theta \) by minimizing (21) with learning rate \( \rho_2 \).
    14. end for
13. **Calculate clustering labels:**
    15. for \( x_i \) in \( X \) do
       16. Calculate the clustering label based on \( f_0(x_i) \) and (19).
    17. end for

---

**IV. EXPERIMENTS**

In this section, we conduct a series of experiments to verify the effectiveness of our clustering algorithm. All the experiments are performed on a desktop workstation with Inter(R) Core i7-4790 3.6GHz CPU, 32G RAM, Ubuntu 14.04 operating system and Keras environment.

### A. DATASETS

We select four representative image datasets including MNIST, Fashion-MNIST, Cifar-10 and STL-10 datasets, to verify the effectiveness of our algorithm. Next, we briefly introduce these datasets.

1) **MNIST AND FASHION-MNIST**

MNIST is a handwritten digits database which includes a training set of 60,000 examples and a test set of 10,000 examples. Fashion-MNIST is a dataset of Zalando’s article images which is an update version of MNIST [38]. For these two dataset, each example is a 28 × 28 gray scale image, associated with a label from 10 classes.

2) **CIFAR-10**

CIFAR-10 contains 50,000 training images and 10,000 test images from 10 classes [39]. Each image has a size of 32 × 32.
3) STL10
The STL-10 dataset is an image dataset used to develop unsupervised feature learning, deep learning and self-supervised learning algorithms [40]. It is inspired by the CIFAR-10 dataset but with some modifications. The high-resolution dataset (96 \times 96) will make it a challenging benchmark to develop more scalable unsupervised learning methods. The detailed statistics of these four datasets are shown in Table 1.

| Datasets      | Samples | Classes | Dimensions |
|---------------|---------|---------|------------|
| MNIST         | 70000   | 10      | 28 \times 28 |
| Fashion-MNIST | 70000   | 10      | 28 \times 28 |
| CIFAR-10      | 60000   | 10      | 32 \times 32 |
| STL-10        | 13000   | 10      | 96 \times 96 |

### B. EVALUATION METRICS
To evaluate the performance of the clustering algorithms, we adopt three commonly metrics including clustering accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). These three metrics reflect the cluster performance from different perspectives. ACC measures the best matching between unsupervised clustering results and ground truth. NMI measures the similarity between pairs of clusters [41], [42]. Adjusted Rand Index (ARI) establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model [43]. All the metrics are the higher the score, the better the clustering performance.

### C. COMPETITORS
We compare the performance to traditional clustering methods, two-stage clustering methods and one-stage clustering methods. Specifically, traditional clustering methods include K-means++ [4], Self-tuning Spectral Clustering (SSC) [6] and Density Based Spatial Clustering of Applications with Noise (DBSCAN) [44]. These methods first extract BOW feature and then cluster the image feature. Two-stage clustering methods include Greedy Layer-Wise Training of Deep Networks (GLWTDN) [45], Deep Embedding Clustering (DEC) [25] and Improved Deep Embedding Clustering (IDEC) [46]. These methods first train an Autoencoder and then use clustering methods to calculate clustering assignments. One-stage clustering methods include Gaussian Mixture Variational AutoEncoders (GMVAE) [22], Categorical Generative Adversarial Networks (CatGAN) [19] and Deep Adaptive Clustering (DAC) algorithm [24]. These three methods also belong to clustering algorithm based on generation model.

1) TRADITIONAL IMAGE CLUSTERING METHODS
K-means++, SSC and DBSCAN: These three comparisons first use BOF algorithm to encode the images. Then, the image features are clustered to achieve image clustering.

2) TWO-STAGE DEEP CLUSTERING METHODS
GLWTDN: It first trains an AE to extract image features, and then uses k-means algorithm to cluster the images features to realize image cluster [45].

DEC: It first learns image representations from an AE. Then, cluster are obtained by utilizing a self-training mechanism [25].

GMVAE: GMVAE uses gaussian mixture model as a prior distribution to improve the traditional variational autoencoder. It uses the improved latent vector as image representation, and then clusters representation to realize image clustering [22].

3) ONE-STAGE DEEP CLUSTERING METHODS
IDEC: IDEC is an improved version of DEC. It trains AE’s reconstruction loss function and self-training loss function simultaneously to guarantee local structure preservation [46].

CatGAN: It uses general Generative Network Adversarial (GAN) and entropy as loss function to realize image clustering [19].

DAC: It formulates image clustering as a binary pairwise classification problem, and identifies this pairs of images which should belong to the same cluster [24].

### D. EXPERIMENT SETUP
For the traditional clustering algorithms, i.e., K-means, SSC and DBSCAN, we first extract the BOF features of the images. The selected image feature extraction method is Scale-Invariant Feature Transform (SIFT) [47], and the number of bins in BOF algorithm is set to 20. The parameters of other comparison methods are mostly set according to the original literature. For our algorithm, we set the parameters $\alpha = 0.01$, $\beta = 0.5$, $\gamma = 0.5$, respectively. According to most of methods based on self-training scheme, the parameter of the student’s t-distribution is set to $v = 1$. We set the learning rates as $\eta = 0.005$ and $\rho = 0.0001$, which are set empirically.
TABLE 2. The network architecture of different databases in the experiment.

| Dataset         | MNIST and Fashion-MNIST dataset                                      | CIFAR-10 dataset          | STL-10 dataset         |
|-----------------|---------------------------------------------------------------------|----------------------------|-------------------------|
| $f_0(x; z_g)$   | Conv. [3 x 3], stride=2, 8 BN Relu                                 | Conv. [3 x 3], stride=4, 16 BN Relu | Conv. [3 x 3], stride=5, 8 BN Relu |
| $g_0(x; z_g)$   | Fc. (dim$_x$ + dim$_z$), BN Relu                                   | Fc. (dim$_x$ + dim$_z$), BN Relu | Fc. (dim$_x$ + dim$_z$), BN Relu |
| $g_0(x; z_g)$   | Fc. (dim$_x$, 256), BN Relu                                        | Fc. (dim$_x$, 256), BN Relu | Fc. (dim$_x$, 256), BN Relu |
| S               | Fc. (dim$_x$, 256), BN Relu                                        | Fc. (dim$_x$, 256), BN Relu | Fc. (dim$_x$, 256), BN Relu |

TABLE 3. Clustering performance and comparison, ACC (%) and NMI (%) and ARI (%), on all datasets. The results marked † are excerpted from [24], [25] and [46]. The best and second best results are highlighted in bold and underlined, respectively.

| Datasets | MNIST (NMI | ARI | ACC | NMI | ARI | ACC | CIFAR-10 (NMI | ARI | ACC | STL-10 (NMI | ARI | ACC |
|----------|------------|-----|-----|-----|-----|-----|----------------|-----|-----|----------------|-----|-----|
| K-means++ [4] | 55.35 | 56.58 | 60.21 | 18.55 | 18.37 | 24.89 | 19.36 | 17.98 | 23.57 | 12.21 | 14.01 | 18.54 |
| SSC [6] | 66.26 | 56.47 | 69.69 | 18.35 | 18.23 | 24.81 | 18.77 | 18.92 | 24.48 | 13.88 | 15.52 | 18.88 |
| DBSCAN [44] | 72.72 | 69.81 | 77.52 | 20.31 | 18.72 | 25.74 | 19.98 | 30.11 | 25.53 | 18.47 | 17.94 | 25.33 |
| GMVAE [22] | 86.55 | 83.72 | 88.56 | 25.68 | 20.71 | 30.10 | 23.93 | 16.89 | 31.35 | 20.60 | 18.61 | 27.90 |
| DEC [25] | 81.60 † | 84.11 | 84.41 | 45.68 | 46.07 | 54.65 | 25.68 † | 16.07 † | 30.10 † | 27.60 † | 18.61 † | 35.90 † |
| IDEC [46] | – | 86.72 † | 88.06 † | 46.31 | 47.11 | 56.27 | 26.31 | 17.11 | 32.24 | 30.26 | 20.33 | 37.35 |
| CatGAN [19] | 79.64 | 76.70 | 88.54 | 28.47 | 18.37 | 33.64 | 34.65 | 26.72 | 45.84 | 23.65 | 15.72 | 21.74 |
| DAC [24] | 93.51 † | 94.86 † | 97.75 † | 43.74 | 39.99 | 47.78 | 30.99 | 30.59 | 52.18 † | 36.55 | 26.65 | 46.91 † |
| Ours | 91.77 | 90.81 | 95.56 | 59.38 | 57.41 | 65.52 | 43.17 | 40.12 | 55.67 | 45.31 | 42.88 | 50.16 |

FIGURE 3. The t-SNE visualization of image representation for Fashion-MNIST dataset.

The detailed network architectures are shown in Table 2. The weights of convolutional and fully connected layers are all initialized by Xavier approach [48].

E. RESULTS

1) CLUSTERING PERFORMANCE COMPARISON

In this part, we compare our method with many state-of-the-art methods including K-means ++ [4], SSC [6], DBSCAN [44], GLWTDN [45], DEC [25], IDEC [46], GMVAE [22], CatGAN [19] and DAC [24]. For our method, we followed the implementation details and report the average results from 5 trails. For the rest, we present the best reported results either from experiment on the original codes of their papers, or from [24], [25] and [46]. We report the detailed clustering results of these methods on all the datasets in Table 3.
FIGURE 4. The t-SNE visualization of image representation for STL-10 dataset.

As shown in Table 3, for each dataset, the performances of deep clustering algorithms are better than that of traditional clustering algorithms. Our clustering method outperforms traditional algorithm with a large margin, which shows the fascinating potential of the proposed method in clustering tasks. Furthermore, note that the proposed method outperforms the deep clustering methods on all the three evaluation metrics expect on MNIST dataset. Our algorithm outperforms all competitive baselines, with significant margins of 7.52%, 3.49% and 3.17% in the case of Fashion-MNIST, Cifar10 and STL10 respectively. These results show the effectiveness of our method in image clustering tasks.

Fig. 2 shows the confusion matrixes of the clustering results for all the datasets. The values along the diagonal represent the percentage of samples correctly classified into the corresponding categories. We can find that all the clustering accuracies are average and stable for all the datasets. This proves that our method does not aggregate samples into a few categories or assign a cluster to outlier samples, and can effectively avoid degenerate solutions problem.

2) VISUALIZATION

In this part, we use two methods to visualize the clustering results of our algorithm. In the first visualization experiment, we map the image representation \( Z \) to a 2-dimension vector by using t-SNE algorithm [49]. We report the t-SNE results of Fashion-MNIST dataset and STL-10 dataset with different clustering accuracy in Fig. 3 and Fig. 4. Different colors indicate different clusters and the corresponding clustering accuracies are reported below. The visualization results show that the proposed algorithm can effectively improve the sep-

FIGURE 5. Qualitative analysis of clustering results of Fashion-MNIST and CIFAR-10 datasets.
arability of data, which is helpful to improve the clustering accuracy.

In the second visualization experiment, we qualitatively analysis the cluster results by the proposed method on Fashion-MNIST dataset and CIFAR-10 dataset. For each category, we randomly select one image as the original image at the first stage. Then, we pick up 5 samples which are the smallest Euclidean distance between original image from the same cluster. Finally, we pick up 5 samples which are most closest to the original image in the incorrect clustering images. All the picked images are shown in Fig. 5, and we mark the correct samples and the incorrect samples with green labels and red labels, respectively. Form the visualization results we can find that the successful cases not only depend on appearance textures, but also contain some semantic information of categories. The failure cases also contain a lot of texture contents similar to the source images. The visualization results imply that our method not only captures image appearance information, but also captures some abstract image information for image clustering. This is the reason why the proposed method can precisely discover the categories of the input images.

3) ON EFFECT OF THE NUMBER OF CLUSTERS

In this experiment, we mainly study the effect of the number of clusters on our algorithm. For each dataset, we conduct 6 experiments on different training sets. The number of training sets varies in the range of [5, 10] at equal intervals. We report the variation curves of clustering accuracy with the number of clusters in Fig. 6.

As shown in Fig. 6, with the increase of the number of clusters, the accuracies of all the clustering methods decrease gradually. For the Fashion-MNIST, CIFAR-10 and STL-10 datasets, the clustering accuracy of our method is always higher than the other algorithms in different number of clusters. In addition, other two metrics results also show the superiority of the proposed algorithm. This is because our algorithm can exploit the essential information of images. In addition, the experimental results also show the stability of the our algorithm.

V. CONCLUSION

This paper proposes a new image clustering method based on Deep Mutual Information Maximization (DMIM) system and self-training algorithm. To make the learned image representation contains more image information, we first derive a deep mutual information maximization system, and use it to learn an unsupervised image representation. To discover the image clusters assignments, we borrow a self-training mechanism and incorporate to the learning of image representation. We evaluate our method on unsupervised clustering tasks using popular datasets, achieving competitive results compared to the current state of the art methods.

Form the view of learning scheme, this paper regards an unsupervised learning problem as a semi-supervised learning problem with enhance the high confidence pseudo labels. Future work may include exploring more self-training methods to assist encoders in obtaining image representations. Specifically, we first train a classification DMIM model and select high confidence pseudo labels as initial labels. Then, we may use some semi-supervised leaning schemes to train the model. Nevertheless, how to determine the initial label is an open problem. A optional way is to automatically select the initial labels based on more prior information. Beside, Graph Convolutional Network (GCN) has been proved to be effective in semi-supervised classification tasks [50]–[52]. A possible direction is adding some GCN layers to the proposed model to improve the clustering performance.

REFERENCES

[1] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[2] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, “Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5736–5745.
[3] B. Dong and X. Wang, “A transfer learning based Feature-Weak-Relevant method for image clustering,” 2018, arXiv:1808.04068. [Online]. Available: http://arxiv.org/abs/1808.04068.
[4] D. Arthur and S. Vassilvitskii, “K-means++, The advantages of careful seeding,” in Proc. 10th Annu. ACM SIAM Symp. Discrete Algorithms Soc. Ind. Appl. Math., 2007, vol. 11, no. 6, pp. 1027–1035.
[5] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, “Laplacian regularized Gaussian mixture model for data clustering,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 9, pp. 1406–1418, Sep. 2011.
[6] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 1601–1608.
[7] F. Moonsmann, B. Triggs, and F. Jurie, “Fast discriminative visual codebooks using randomized clustering forests,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 985–992.
[8] N. Ahmed, “Recent review on image clustering,” IET Image Process., vol. 9, no. 11, pp. 1020–1032, Nov. 2015.
[9] J. E. Kaya, A. Ç. Pehlivan, E. G. Sekizkardes, and T. Birci, “PCA based clustering for brain tumor segmentation of T1w MRI images,” Comput. Methods Programs Biomed., vol. 140, pp. 19–28, Mar. 2017.
[10] W. Zhang, X. Wu, W.-P. Zhu, and L. Yu, “Unsupervised image clustering with SIFT-based soft-matching affinity propagation,” IEEE Signal Process. Lett., vol. 24, no. 4, pp. 461–464, Apr. 2017.
[11] D. Hong, N. Yokoya, N. Ge, J. Chausssot, and X. X. Zhu, “Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification,” ISPRS J. Photogramm. Remote Sens., vol. 147, pp. 193–205, Jan. 2019.
[12] D. Hong, N. Yokoya, J. Chausssot, and X. X. Zhu, “An augmented linear mixing model to address spectral variability for hyperspectral unmixing,” IEEE Trans. Image Process., vol. 28, no. 4, pp. 1923–1938, Apr. 2019.
[13] T. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu. J. Zhang, M. Ester, and C. Wang, “ANRL: Attributed network representation learning via deep neural networks,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 3155–3161.
[14] N. Tishby, “The information bottleneck theory of deep neural networks,” APS, vol. 2018, p. K58, 2018.
[15] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks,” Synth. Lectures Comput. Archit., vol. 15, no. 2, pp. 1–341, 2020.
[16] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, “A survey of clustering with deep learning: From the perspective of network architecture,” IEEE Access, vol. 6, pp. 39501–39514, 2018.
[17] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, “Clustering with deep learning: Taxonomy and new methods,” 2018, arXiv:1801.07648. [Online]. Available: http://arxiv.org/abs/1801.07648.
[18] K.-L. Lim, X. Jiang, and C. Yi, “Deep clustering with variational autoencoder,” IEEE Signal Process. Lett., vol. 27, pp. 231–235, 2020.
A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks," in *Proc. Adv. Neural Inf. Process. Syst.* (ICINOP). Charn, Switzerland: Springer, 2017, pp. 373–382.

J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep unsupervised learning with consistent inference of latent representations," *Pattern Recognit.*, vol. 77, pp. 438–453, May 2018.

N. Didokhanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," 2016, arXiv:1611.02649. [Online]. Available: http://arxiv.org/abs/1611.02640.

J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5174–5176.

J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 5879–5887.

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

X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 21, 2019, doi: 10.1109/TNNSL.2019.2958324.

K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised Spectral–Spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, Dec. 2015.

X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

A. Hayat, U. Singh, and V. P. Namboodiri, "InfoRL: Interpretable reinforcement learning using information maximization," 2019, arXiv:1905.10404. [Online]. Available: http://arxiv.org/abs/1905.10404.

W. Li and Y. Li, "Entropy, mutual information, and systematic measures of structured spiking neural networks," *J. Theor. Biol.*, vol. 501, Sep. 2020, Art. no. 110310.

D. Xiao, C. Qin, H. Yu, Y. Huang, and C. Liu, "Unsupervised deep representation learning for motor fault diagnosis by mutual information maximization," *J. Intell. Manuf.*, pp. 1–15, Apr. 2020.

R. Devon Hjeltn, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," 2018, arXiv:1808.06670. [Online]. Available: http://arxiv.org/abs/1808.06670.

S. Nowozin, B. Cseke, and R. Tomioka, "FGAN: Training generative neural samplers using variational divergence minimization," 2016, arXiv:1606.00709. [Online]. Available: http://arxiv.org/abs/1606.00709.

G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov et al.’s negative-sampling word-embedding method," 2014, arXiv:1402.3722. [Online]. Available: http://arxiv.org/abs/1402.3722.

J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 355–370, Feb. 2017.

H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, arXiv:1708.07747. [Online]. Available: http://arxiv.org/abs/1708.07747.

A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2011, pp. 215–225.

D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

S. Yu received the B.Sc. and M.Sc. degrees in pattern recognition and intelligent system from Liaoning Shihua University, China, in 2011 and 2015, respectively. He is currently pursuing the joint Ph.D. degree with Northeastern University, China, and the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include image/video representation, unsupervised learning, and deep learning.

Siquan Yu

Guiping Shen received the B.Sc. degree in automation from Anhui University, China, in 2009, and the M.Sc. and Ph.D. degrees from the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China, in 2012 and 2019, respectively. She is currently an Associate Professor with the College of Artificial Intelligence, Yangzhou University, Fuzhou, China. Her current research interests are machine learning and machine vision.
PEIYAO WANG received the B.Sc. and M.Sc. degrees in pattern recognition and intelligent system from Liaoning Shihua University, China, in 2014 and 2017, respectively. She is currently a Teaching Section Chief with the Shenyang Institute of Technology. Her research interests include image/video representation and deep learning.

YUNING WANG received the B.Sc. degree in ammunition engineering and explosive technology from the College of Equipment Engineering, Shenyang Ligong University, Shenyang, Liaoning, China. He is currently working as an Assistant Engineer with PLA 32681. His research interests include the theories and algorithms of object detection, machine learning, and intelligent vision systems.

CHENGDONG WU (Member, IEEE) is currently the Vice President of the Faculty of Robot Science and Engineering, Northeastern University, and the Director of the Institute of Artificial Intelligence, a Professor, and a Doctoral Tutor with Northeastern University, Shenyang, China. He has long been involved in automation engineering, artificial intelligence, and teaching and researching in robot navigation. He is also an Expert in Chinese modern artificial intelligence and robot navigation. He is also a Special Allowance of the State Council.

* * *