Cross-Domain Contrastive Learning for Hyperspectral Image Classification

Peiyan Guan, *Graduate Student Member, IEEE*, and Edmund Y. Lam, *Fellow, IEEE*

***Abstract*— Despite the success of deep learning algorithms in hyperspectral image (HSI) classification, most deep learning models require a large amount of labeled data to optimize the numerous parameters. However, it is very expensive and time-consuming to collect a lot of labeled HSI samples. To cope with this problem, we propose a cross-domain contrastive learning (XDCL) framework to learn representations of HSIs in an unsupervised manner. We demonstrate that the features that are valuable for category identification are shared across the spectral and spatial domains, while the less useful contents tend to be independent. The XDCL extracts such domain-invariant information with a cross-domain discrimination task, i.e., predicting which two representations of different domains are matched. With this insight, our method learns semantically meaningful HSI representations. We develop a simple method to construct effective signals representing the two domains, respectively. Moreover, we randomly mask the signals to improve their semantic level and encourage the representations to dig out more useful abstract factors. In order to evaluate the representation quality, we use the learned representations to train a linear classifier on three hyperspectral datasets with limited labeled samples. Experimental results demonstrate that our method surpasses the state-of-the-art methods by a large margin.**

***Index Terms*— Contrastive learning, hyperspectral image classification, limited labeled samples.**

I. Introduction

HYPERSPECTRAL imaging technology is able to capture images of the ground objects with hundreds of narrow spectral bands covering a large wavelength range. [1]. It has been applied in various areas of Earth observation, such as environmental monitoring [2], [3], mineral exploration [4], and landcover classification [5]. HSI classification, which aims to label each pixel of the image, is a crucial analysis technique used in applications. Since classifying the raw spectrum has very poor performance, researchers propose to extract discriminative features for better classification. Earlier work mainly focuses on spectral features extraction [6]–[9]. Later, some methods propose to extract the spatial and spectral features jointly [10]–[12]. Some traditional methods make use of the label information to extract effective features, where the samples of different classes are separated. Representative methods include nonparametric weighted feature extraction (NWFE) [13], linear discriminant analysis (LDA) [14], and many variants of these two [15]. In contrast, some other methods are designed to learn the representations without labels. One typical method is the principal component analysis (PCA) [7]. It is used to reduce the dimensionality of HSIs while preserving as much information as possible. Besides, various manifold learning approaches have been proposed to extract features by discovering the manifolds embedded in the high-dimensional spaces [16], [17]. These methods make use of strong prior knowledge and usually have a small number of parameters to be tuned. However, they can only extract shallow handcrafted features, which have the weak discriminative ability.

Recently, deep learning has achieved great success in many image processing applications [18]–[20]. Deep networks, such as the convolution neural network (CNN), show a strong capability of extracting high-level features for pattern recognition [21]–[23]. Motivated by such success, a lot of works have applied deep neural networks in HSI classification [24]–[26] and demonstrated superior performance when a lot of labeled data are available, which are called supervised learning methods. Representative methods for spectral feature extraction include 1-D CNN [24], the recurrent neural network (RNN) [27], and the deep belief network (DBN) [28]. To integrate the spatial information and spectral information for more accurate classification, 2-D and 3-D CNNs have also been applied to extract deep representations of the HSIs [25], [26]. In addition, many variants of these networks with stronger feature extraction capability have been proposed for more accurate classification. For example, Zhang *et al.* propose a multiscale dense network to utilize useful information of various scales for feature extraction [29]. Zhu *et al.* design a residual spectral–spatial attention network (RSSAN) with stacked residual modules, which are incorporated with spectral and spatial attention mechanisms, to learn effective representations from HSIs [30]. In spite of the impressive performance of these networks in feature extraction and classification, a large amount of labeled data is required to optimize their numerous parameter. However, it is time-consuming and may even be infeasible to collect sufficient labeled HSI samples.

To address this issue, a variety of approaches have been proposed. Deep unsupervised learning, which makes use of abundant and accessible unlabeled data to extract deep features, is one of the major classes. Conventional unsupervised learning methods employ strong models, such as autoencoder

Manuscript received January 23, 2022; revised May 6, 2022; accepted May 15, 2022. Date of publication May 20, 2022; date of current version June 1, 2022. This work was supported in part by the Research Grants Council of Hong Kong (GRF) under Grant 17200019, Grant 17201620, and Grant 17200321 and in part by the University of Hong Kong under Grant 104005864. *(Corresponding author: Peiyan Guan.)*

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: pyguan@eee.hku.hk; elam@eee.hku.hk).

Digital Object Identifier 10.1109/TGRS.2022.3176637

(AE), to learn deep representations by reconstructing the input data completely [31], [32]. These representations are thus pushed to compress all the contents of the original inputs. However, HSIs usually contain massive noise and heavy redundancies, which are actually useless and might be bad for identifying the target. Such lossless representations are, thus, not quite suitable for classification. On the contrary, we aim at learning deep representations that only encode the contents valuable for classification. Thus, the question becomes: how can we differentiate the desired contents versus the undesired ones without labels?

Let us take a look at the hyperspectral data first. An HSI captures the object information of both spectral and spatial domains with the spectrum and spatial images, respectively [33]. Specifically, the former presents the inherent spectral reflectance property, while the latter exhibits the shape, texture, structure, and neighborhood relationship of the object. Thus, the spectrum of all pixels and the spatial images of all bands in one HSI can be considered as perceiving the same scene from different perspectives. In other words, despite containing information from different domains, they share the same semantics. This is significant evidence that the good contents useful for object identification are encoded in the information shared between the two domains, while the undesired ones, such as noise and redundancy, are usually independent. The shared information helps us filter out the nuisance information and keep the good ones naturally. We argue that encoding the shared information of different domains is a better choice to learn powerful HSI representations. Therefore, our objective is to discover and extract the information shared between the two domains.

To achieve this goal, we propose a simple and effective method, called cross-domain contrastive learning (XDCL), to learn HSI representations in an unsupervised manner. The idea behind the XDCL framework is to contrast the two domains. Specifically, we design a cross-domain discrimination task that predicts which two representations of different domains belong to the same sample. This task pushes the matched representations closer, while those of different samples are far apart. The representations are, thus, able to capture the information shared between the two domains. We show that constructing a pair of strong signals representing the two domains is critically important. To make the representations further suitable for classification, we propose to randomly mask the constructed signals. Such masking enables the representations to focus less on low-level contents and code the desired information in terms of more high-level factors. By performing linear classification on the learned representations with few labels over various HSI datasets, we show that encoding the shared information of the two domains only is more powerful than compressing the whole information for object identification, which demonstrates the superiority of our method over the state-of-the-art methods. Our main contributions can be summarized as follows.

1) To the best of our knowledge, we are the first to investigate the information shared across the spectral and spatial domains for the unsupervised learning of HSI representations.
2) We propose an XDCL framework to learn HSI representations suitable for classification by encoding only the information shared between the two domains.
3) We develop a simple method to construct pairs of powerful visual and spectral signals representing the two domains, respectively, and give a detailed analysis of the effect of our strategy.
4) We push the representations further toward encoding the shared information in terms of high-level factors instead of shallow contents by masking parts of the signals randomly.

The rest of this article is organized as follows. Section II introduces the related work on HSI classification with limited labeled data. Section III gives detailed description of the XDCL. Section IV presents the experimental results on collected datasets and some discussion about our method. Conclusions are drawn in Section V.

II. Related Work

Current deep learning methods usually build deeper, denser, or higher dimensional networks to obtain stronger feature extraction capability [34], [35]. However, they are more likely to encounter overfitting problems with limited labels. To overcome this, a variety of methods have been proposed for HSI classification. We give a brief review of related work in this section.

A. Unsupervised Learning

It aims to capture high-level features with plenty of unlabeled data and then conduct classification on the learned features. AE is one of the most commonly used structures for the unsupervised feature extraction. It learns a latent representation by trying to reconstruct the original input data completely [36]. Many unsupervised learning approaches based on the AE have been developed for HSI classification [37]. One early example is the deep AE (DAE), which employs a 1-D AE to learn the latent representation of the input vectors [31]. Mei *et al.* builds a 3-D convolutional AE (3-DCAE) to extract the spectral–spatial features more effectively [32]. Besides, the generative adversarial network (GAN) has also been widely applied to learn HSI representations in an unsupervised fashion [38], [39]. Unlike the AE, the GAN model employs an additional discriminator to guide the generator to learn the distribution of real data, while the discriminator is pushed to learn high-level features of the HSI samples. These models are designed to learn representations by pixelwise reconstruction or generation. Nevertheless, the HSI, as a natural signal, is very low semantic and has heavy spatial and spectral redundancies. Encoding such contents makes the representations perform very poorly on classification tasks. Recently, self-supervised learning has been introduced to the classification of HSI with limited labeled samples [40]. These methods encourage the representations to distinguish more useful information without labels by designing some pretext tasks.

B. Semisupervised Learning

It uses few labeled and abundant unlabeled data together to enhance the feature extraction and classification procedures simultaneously. Some of them propose to train the feature extractor with unlabeled samples and then fine-tune the extractor and classifier jointly with the labeled samples [41], [42]. The main disadvantage of these methods is that the network faces potential overfitting risk, while the number of labeled samples is too small. Some other approaches predict pseudolabels for the unlabeled data, which are used to extend the labeled dataset and are then used to train the feature extractor and the classifier [43]. Plenty of samples with high-confidence pseudolabel enrich the dataset greatly and improve the generality of the network. However, one common drawback of these methods is that one small prediction error may accumulate and result in the final collapse of the network training.

C. Few-Shot Learning

It is a class of methods specifically designed to distinguish new categories with very few labeled samples [44]. It is first proposed for image classification in computer vision. Unlike conventional deep learning models, few-shot learning algorithms aim at learning how to learn, also known as metalearning. Recently, many researchers have introduced it into the HSI classification problem. Typical methods include the model-agnostic metalearning algorithm [45], the prototypical network [46], and the relation network [47]. These methods learn how to learn by training the models on a well-labeled dataset of relevant domains and transfer the trained model to the target dataset with few labeled samples. Thus, it can be considered a special type of transfer learning [48]. However, these methods have high demands for the amount and diversity of the labeled source dataset. Besides, the source domain is required to be related to the target domain as a guarantee of good transfer performance.

D. Contrastive Learning

This technique has become one of the most competitive methods for learning representations without labels [49]. It builds a contrastive loss to push the similar instances closer while pulling the dissimilar ones apart. It has been widely used to learn a variety of data representations, such as image, text, and video [50], [51]. One key point in designing a contrastive learning algorithm lies in how to select similar instances without supervision. The most common method is to create multiple views of each data. Examples include the chrominance and luminance of an image [52], multiple augmentations of the same data [49], different clips of a video [53], or sound and video [51]. The dissimilar instances can be randomly selected from different data. In our work, we create different views of an HSI sample by constructing a pair of signals representing the spatial and spectral domains, respectively. Our method seeks to learn strong HSI representations by finding the information shared between the two domains, which is useful for object identification.

III. Proposed Method

A. Overview of the Framework

Consider that we have a collection of N unlabeled HSI samples $\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{H\times H\times C}$ as the dataset, where H and C denote the patch size and the band number of the hyperspectral cubes. Each sample is considered to be the same as itself only and different from all the others. The objective of our method is to learn powerful representations of these samples in an unsupervised manner. As discussed before, extracting the information shared between the spectral and spatial domains helps us separate the useful contents from the uninformative ones. The XDCL learns representations encoding such good information shared between different domains via the cross-domain discrimination task. The framework of the XDCL is presented in Fig. 1.

As can be seen, it consists of four major steps. First, given an HSI sample, we construct a pair of visual and spectral signals for it to represent its spatial information and spectral information by selecting a spectrum and a band, respectively. The two signals of the same sample are considered a positive pair, while those of different samples are referred to as a negative pair. We use these signal pairs to explore the information shared across the two domains. Then, we generate two masks to remove parts of the two signals randomly with a ratio, which reduces noise and redundancy greatly, thus improving the semantic level of the signals. Afterward, we build two different base encoders, which are implemented with deep CNNs, to extract representations from the two masked signals, respectively. In the end, the cross-domain discrimination task maximizes the agreement between the two representations of the same sample and minimizes the similarity between those of different samples. We define a contrastive loss function based on the noise-contrastive estimation (NCE) [54] for this task. To sum up, the XDCL framework first constructs two signals for each sample, which contains information on the two domains. The objective of our work then turns to extract the shared information of the two signals. Their semantic levels are further enhanced by random masking. Then, our framework employs contrastive learning to learn a feature embedding that separates the signals of different samples and matches those of the same sample. Aligning the signals in embedding spaces enables us to extract the desired semantic features. In this way, the XDCL learns powerful HSI representations that capture the good contents invariant between the spectral and spatial domains.

B. Signal Construction

While we define the cross-domain discrimination task on the visual and spectral signals, the XDCL is actually designed to extract the information invariant across the two signals. Thus, how to construct effective signals is of crucial importance. Learning powerful representations requires the information shared between the two signals to be highly related to semantics. We propose a simple method for signal construction. Specifically, we use the center spectrum of the sample cube as the spectral signal $s \in \mathbb{R}^C$, as it captures the information of the object in the spectral domain. The situation is more

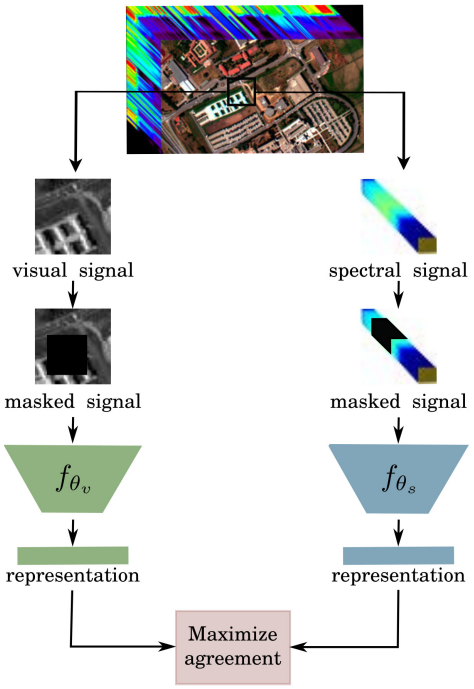

Fig. 1. Architecture of the XDCL framework.

complicated when it comes to the visual signal. Each band of the HSI is a 2-D image, which has no spectral information but partial spatial information of the scene, and different bands contain spatial information at different wavelengths. Thus, every band can be considered a signal, which leaves us with many choices. On the one hand, using a single band does not make full use of the spatial information, which goes against extracting more useful features. On the other hand, using multiple bands introduces undesired spectral information into the visual signal, which probably lowers the representation quality.

According to the two considerations, we propose a straightforward construction strategy: we select a random band, which varies in every epoch of the training procedure, as the visual signal $v \in \mathbb{R}^{H\times H}$. We simply refer to this strategy as "random selection." In this way, more and more spatial information will be covered as the training progresses, while no spectral information will be included. In addition, another advantage of this strategy over using one or multiple fixed bands is that it augments the signals in a natural way and enhances the data diversity, which helps prevent overfitting problems. The two constructed signals contain abundant information about their corresponding domains and avoid potential interdomain mixing. We show that this simple method is able to construct signals with desired shared information.

C. Signal Masking

The XDCL aims to dig out more high-level factors and less shallow (e.g., pixel) contents from the information shared between the signals. A feasible approach is to improve the semantic level of the signals. We consider randomly masking the signals when extracting latent representations. Specifically, we generate a pair of masks $M_v \in \mathbb{R}^{H\times H}$ and $M_s \in \mathbb{R}^C$ to

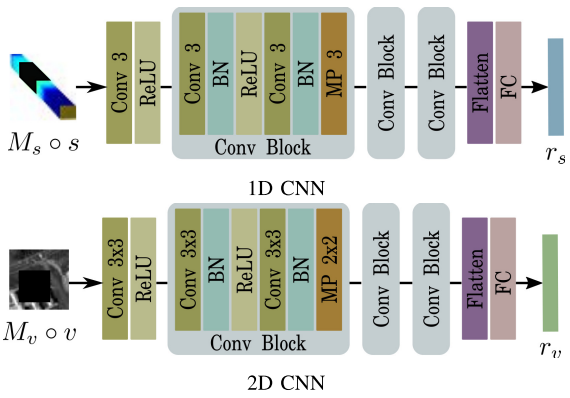

Fig. 2. Architectures of the 1-D and 2-D CNNs. The convolution (Conv) blocks of in both networks consists of two convolution layers, two batch normalization (BN) layers, an ReLu function, and a max pooling (MP) layer. FC represents the FC layer. (a) 1-D CNN. (b) 2-D CNN.

randomly remove blocks, i.e., set to zero, of the signals v and s with a ratio r, respectively. We call this strategy "blockwise masking."

Randomly removing parts of the signal eliminates its noise and redundancy greatly and breaks the extraction of low-level information. However, erasing excess contents might create a representation learning task that cannot be solved easily. Thus, the masking ratio r needs to be set to an appropriate value. In addition, we find that masking the two signals simultaneously results in a higher training loss and makes the network hard to converge. In this case, simply lowering the mask ratio is not the best choice since it would keep more low-level contents and reduce the model's capability of extracting abstract factors. To solve this problem, we consider an asymmetric signal masking strategy where only one of the two signals is randomly selected to be masked at a time, while the other one remains the same (i.e., the corresponding mask is set to 1). Such an asymmetric setting has a similar effect as masking both signals on eliminating low-level shared contents and improves the training speed of the networks. Maximizing the agreement between such masked representations pushes the representations further toward encoding the shared information in terms of abstract factors and is more suitable for object identification.

D. Base Encoders

Two base encoders are built to learn latent representations of the masked signals. The XDCL framework allows many options for the encoder construction. We adopt deep CNNs for performance and simplicity. Specifically, we use a 1-D CNN $f_{\theta_s}(\cdot)$ with parameters θ_s and a 2-D CNN $f_{\theta_v}(\cdot)$ with parameters θ_v to learn the representations as $r_s = f_{\theta_s}(M_s \circ s)$ and $r_v = f_{\theta_v}(M_v \circ v)$, where $\circ$ represents elementwise multiplication, and $r_s, r_v \in \mathbb{R}^D$. The architectures of the two networks are presented in Fig. 2. In both networks, the first convolution layer has 64 kernels, while the following three convolution blocks generate 128, 256, and 512 feature maps, respectively. The feature maps acquired by the last convolution blocks are flattened first and then fed into a fully connected (FC) layer, the input dimensionality of which varies with the size of the

input signals, while the output dimensionality is set to the representation size D.

E. Learning Objective

Our method captures the domain-invariant information by maximizing the agreement between the representations of the positive signal pairs. To avoid the trivial solution where all representations are constant, we also pull those of negative pairs far apart. We adopt a contrastive loss based on the NCE to achieve this. Taking the spectral signals s as the anchor points and enumerating over visual signals v in a minibatch of K samples, the loss can be defined as

$$L_{\mathrm{NCE}}(s,v)=\sum_{i=1}^{K}-\log\frac{\exp\left(\Phi\left(r_s^i,r_v^i\right)/\tau\right)}{\sum_{j=1}^{N}\exp\left(\Phi\left(r_s^i,r_v^j\right)/\tau\right)} \tag{1}$$

where τ represents the temperature hyperparameter and $\Phi(\cdot,\cdot)$ denotes the scoring function, which measures the similarity between representations and has many options, such as the Euclidian distance and the cosine similarity. Since the representations learned by our method are further classified by a linear classifier, which measures the cosine similarity between its weight vectors and the representations, using cosine similarity to align the matched representations in embedding space could also align them in label space. For efficient computation and subsequent classification task, we use the normalized dot product as the scoring function

$$\Phi\left(r_s^i,r_v^i\right)=\frac{\phi_s\left(r_s^i\right)^{\top}\phi_v\left(r_v^i\right)}{\left\|\phi_s\left(r_s^i\right)\right\|\left\|\phi_v\left(r_v^i\right)\right\|} \tag{2}$$

where ϕ_s and ϕ_v denote projections that transform the representations onto some other spaces, which is proven effective for improving the representation quality [55]. We implement the projections with two nonlinear two-layer perceptrons. Symmetrically, we get $L_{\mathrm{NCE}}(v,s)$ by anchoring at visual signals v. We use their sum as the overall contrastive loss

$$L_{\mathrm{NCE}}=L_{\mathrm{NCE}}(s,v)+L_{\mathrm{NCE}}(v,s). \tag{3}$$

Minimizing this loss promotes the representations of the same sample to get much higher similarity scores than those of different samples.

IV. Experimental Results and Discussion

A. Dataset and Evaluation Metrics

We extensively evaluate the performance of the XDCL on three widely used hyperspectral datasets, including Pavia University (PU), Houston, and Kennedy Space Center (KSC). The details of the three datasets are presented as follows.

1) PU: It is captured over the PU by the reflective optics system image spectrometer (ROSIS). It contains nine classes of ground objects. After removing the noisy band, there remain 103 spectral bands covering the spectral range of 430–860 nm, each of which contains 610 × 610 pixels. We only use the top left area with 610 × 340 pixels for our experiment since the other part contains no information.

2) Houston: The image of Houston is captured over the campus and neighborhood area of the University of Houston. After removing the noisy bands, we have an image with a size of 1905 × 349 and 144 spectral bands in the range of 380–1050 nm. 15 classes representing the various land covers are differentiated in this dataset.

3) KSC: It is acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVRIS) sensor during a flight over the KSC. After removing the noisy bands, 176 bands are used in our experiment, which covers the spectral range of 400–2500 nm. The size of each band is 512 × 614. There are in total 13 classes in the dataset.

These datasets cover various ground objects and provide a great diversity of the samples, which enables us to verify the generality of our method. As mentioned before, we select the hyperspectral cubes with a size of $H \times H \times C$ as the input. The class of the input sample is determined by the center pixel. According to our observation, XDCL achieves satisfactory performance when about half the unlabeled samples of the three datasets are used for training and do not show further performance improvement when more samples are used. It demonstrates that using only a part of the samples may be sufficient to capture the information of an HSI dataset. This may be due to the overlap between neighboring samples. Considering that employing more samples increases the training time, in each dataset, 50% unlabeled samples are used to train the XDCL. After the unsupervised training, we fix the parameters of the base encoders and use them as our feature extractors. To verify our method, we then perform linear classification on the representations of labeled data. Specifically, we input the two signals of each labeled sample into the fixed feature extractors and obtain the corresponding representations. The representations of the spectral and visual signals are concatenated to form the full representation of a sample. A linear classifier is then used to classify the frozen representations. Five labeled samples per class are selected to train the classifier, and the remaining ones are used for testing. We evaluate the classification results quantitatively with four metrics, including class-specific accuracy, average accuracy (AA), overall accuracy (OA), and the kappa coefficient (κ). Class-specific accuracy measures how many samples are classified correctly for each class. AA is computed by averaging the accuracy of all the classes. OA assesses the accuracy of all the data. κ evaluates the agreement between the predicted labels and true classes. The higher these four metrics, the better the classification result.

B. Implementation Details

The pixel intensities of the HSIs are normalized to $[0, 1]$. The training procedure is completed after 100 epochs. The XDCL model is trained for 80 epochs over the PU and Houston datasets, and 200 epochs over the KSC dataset. We use the Adam optimizer [56] with $\beta_1 = 0.9$ to optimize the base encoders and classifier, while the learning rate is set to 3×10^{-4} for the base encoders. The temperature hyperparameter is set to 0.07. The XDCL framework is implemented based on Pytorch, and the experiments are conducted on a

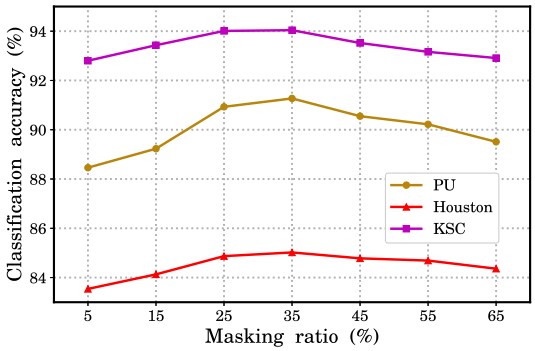

Fig. 3. Overall classification accuracy with different masking ratios of the signal masking over different datasets.

computer platform equipped with a Linux OS, an Intel Core i5-6500 CPU (3.2 GHz, four cores), 16-GB physical memory, and an Nvidia RTX 2060s graphics card.

C. Parameter Setting and Analysis

There are many important parameters that affect the performance of our method. Some of them, such as the learning rate and training epoch, have strong generality over different data, which are thus determined empirically. However, the influence of some other parameters, such as the masking ratio r, the batch size K, and the patch size H, need to be carefully investigated. We conduct classification experiments over the three datasets to analyze them.

1) Masking Ratio: We randomly remove large blocks of the signals to improve their semantic levels with a masking ratio r, which has a great impact on the representation quality. We evaluate its effectiveness quantitatively and present the classification results in Fig. 3. The results of the three datasets follow a similar trend. The OA rises steadily until the sweet point and then falls as the masking ratio increases. Using a lower ratio does not reduce the capture of low-level information shared between the signals, while masking too many parts makes the extraction of semantic features too hard. In both cases, the representation quality is degraded. The optimal ratios for all the datasets lie around 35%, which gives the best classification performance. Thus, we adopt 35% as the default masking ratio in our experiment.

2) Patch Size: By varying the patch size H, we are able to determine how much spatial information is included in the HSI sample, which is important to the visual representation learning. The classification results of different patch sizes are shown in Fig. 4. As can be seen, the OA grows rapidly as the patch size increases from 5 to 9 and presents no significant change while it increases further. It demonstrates that the samples contain insufficient spatial information for the semantic extraction when $H < 9$. Considering that the computational complexity is increased, while bigger patches are used, we adopt 9 as the default patch size in the subsequent experiments.

3) Batch Size: The contrastive loss compares each sample with every other one within a minibatch with a size of K. Using more samples helps us separate those of different classes. The influence of the batch size is presented in Table I.

TABLE I

OVERALL CLASSIFICATION ACCURACY WITH DIFFERENT BATCH SIZES K OVER THE THREE HSI DATASETS. THE BEST VALUE IS IN BOLD

Batch size	PU	Houston	KSC
128	84.55%	80.43%	87.66%
256	88.73%	83.24%	92.18%
512	**91.27%**	**85.02%**	**94.04%**
1024	90.72%	84.65%	93.82%
2048	89.89%	84.13%	92.98%

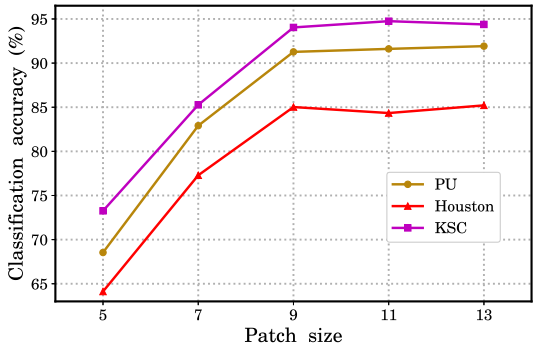

Fig. 4. Overall classification accuracy with different patch sizes H over different HSI datasets.

It is observed that the XDCL gives the best performance over the three datasets when $K = 512$. However, the accuracy is reduced a bit, while more samples are further used. This is probably because more samples belonging to the same class would be considered negative pairs and pushed far apart. Hence, the batch size is set to 512 in the following experiments.

D. Comparison With the State-of-the-Art Methods

In this subsection, we evaluate the XDCL by performing linear classification on the learned representations, while very few labeled samples are available. Specifically, the number of labeled samples is set to 5 per class for three datasets. We compare the XDCL with eight state-of-art methods belonging to different categories that are summarized in five groups as follows.

1) *Traditional Feature Extraction Methods:* Extended morphological profile-support vector machine (EMP-SVM) [10] extracts spectral information with the principal component analysis (PCA) first. Then, it builds the morphological attribute profiles for spatial feature extraction by using morphological filters.
2) *Deep Supervised Learning Methods:* RSSAN [30] propose two modules that incorporate spectral attention and spatial attention, respectively. The dual attention mechanisms are effective for adaptive feature selection [57]. It also inserts the two modules in the conventional residual block for the supervised feature extraction.
3) *Deep Few-Shot Learning Methods:* Relation network for HSI few-shot classification (RN-FSC) [47] contains an embedding module and a relation module. It aims to learn a deep metric space that minimizes the intraclass

TABLE II

CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE PU DATASET. THE BEST CLASS-SPECIFIC, OA, AA, AND KAPPA VALUE κ ARE IN BOLD

Class No.	EMP-SVM [10]	RSSAN [30]	RN-FSC [47]	SS-CNN [41]	AROC-DPNet [43]	3DCAE [32]	WGAN-GP [39]	DMVL [40]	XDCL
1	68.64	69.11	66.33	64.60	77.94	58.60	44.08	77.04	**79.62**
2	53.54	61.34	68.69	61.29	93.26	63.69	75.97	77.19	**95.19**
3	61.72	94.59	95.21	**97.73**	84.91	71.41	92.74	93.56	85.27
4	91.96	86.60	97.35	98.16	96.32	95.71	98.09	91.86	**99.16**
5	99.55	99.85	**100.00**	99.55	98.81	**100.00**	**100.00**	**100.00**	99.78
6	86.22	71.96	80.12	72.32	43.21	76.20	81.43	73.99	**91.57**
7	**99.77**	38.50	98.57	96.02	98.35	98.80	99.55	98.27	95.94
8	57.99	10.76	66.35	48.32	**95.46**	90.39	54.21	80.12	83.60
9	98.94	94.72	98.94	99.89	99.58	**100.00**	99.79	99.79	95.88
OA(%)	67.63	64.03	75.60	69.74	85.21	72.76	74.24	80.86	**91.27**
AA(%)	79.82	69.71	85.73	81.99	87.54	83.87	82.87	87.98	**91.78**
κ(%)	61.22	56.04	69.73	63.28	80.63	67.00	68.05	75.87	**88.69**

TABLE III

CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE HOUSTON DATASET. THE BEST CLASS-SPECIFIC, OA, AA, AND KAPPA VALUE κ ARE IN BOLD

Class No.	EMP-SVM [10]	RSSAN [30]	RN-FSC [47]	SS-CNN [41]	AROC-DPNet [43]	3DCAE [32]	WGAN-GP [39]	DMVL [40]	XDCL
1	40.68	25.62	70.23	25.40	76.42	34.28	31.59	72.85	**82.02**
2	13.89	**81.98**	40.51	49.04	57.15	63.27	72.22	60.80	76.62
3	**99.87**	65.54	98.87	69.06	97.99	78.49	97.36	97.74	98.99
4	93.12	62.74	**93.83**	88.77	89.72	92.01	93.35	91.30	92.64
5	80.66	**96.07**	84.75	71.42	65.87	75.96	87.06	78.43	89.83
6	87.32	85.84	82.89	87.32	89.09	98.23	87.91	89.68	**98.53**
7	47.61	32.20	71.11	46.65	90.16	73.77	76.03	74.80	**94.60**
8	19.75	28.73	38.09	43.36	**52.41**	43.50	35.64	32.67	31.48
9	68.95	84.01	79.24	85.77	64.75	78.72	61.19	**92.89**	88.55
10	33.29	36.80	59.13	78.51	83.92	87.85	67.56	87.99	**98.88**
11	56.15	18.07	74.36	28.46	**96.80**	70.71	77.69	87.31	96.67
12	30.65	47.59	49.34	32.75	**75.30**	41.85	38.42	26.94	65.85
13	95.89	47.31	94.30	92.88	94.46	92.09	93.83	89.24	**96.68**
14	**100.00**	83.24	99.42	97.47	99.03	99.42	95.52	98.64	**100.00**
15	98.25	89.22	98.62	88.35	96.74	98.50	99.12	99.25	**99.88**
OA(%)	56.83	54.96	71.28	60.17	79.11	70.84	69.63	75.25	**85.02**
AA(%)	64.41	59.00	75.64	65.68	81.99	75.24	74.30	78.70	**87.41**
κ(%)	53.53	51.29	69.04	56.93	77.44	68.53	67.19	73.25	**83.82**

distance, maximizes the interclass distance on a source domain, and then transfers the model to the target domain with few labeled samples.

4) *Deep SS Learning Methods:* Semi-supervised (SS)-CNN [41] employs an AE to learn the representations. It combines the supervised cost for labeled samples and the unsupervised cost for unlabeled samples to optimize the network. Approximate rank-order clustering-depthwise and pointwise convolution network (AROC-DPNet) [43] combines the CNN with a clustering algorithm for HSI classification. It alternately optimizes the network and predicts pseudolabels for the unlabeled data, which are then used to extend the training dataset.

5) *Deep Unsupervised Learning Methods:* 3-DCAE [32] develops a 3-D convolutional AE to learn unsupervised representations. The 3-D convolution extracts joint spectral–spatial features effectively. Wasserstein GAN with gradient penalty (WGAN-GP) [39] builds a generative adversarial model that consists of a generator and a discriminator. The competition between the two modules pushes the generator to learn the data distribution and the discriminator to have the powerful capability of feature extraction. Deep multiview learning (DMVL) [40] employs a deep residual network to learn the deep representation. Residual learning eases the training of deeper networks [58], [59]. DMVL divides all the bands of HSI samples into two groups and designs a pretext

TABLE IV

CLASSIFICATION RESULTS OF DIFFERENT METHODS ON THE KSC DATASET. THE BEST CLASS-SPECIFIC, OA, AA, AND KAPPA VALUE κ ARE IN BOLD

Class No.	EMP-SVM [10]	RSSAN [30]	RN-FSC [47]	SS-CNN [41]	AROC-DPNet [43]	3DCAE [32]	WGAN-GP [39]	DMVL [40]	XDCL
1	82.31	**99.58**	93.18	97.35	87.88	90.67	54.74	92.62	94.85
2	70.78	10.29	64.61	82.31	59.67	70.78	68.31	72.43	**91.77**
3	60.16	88.67	93.36	92.97	97.27	64.45	**98.05**	83.98	97.66
4	64.68	7.94	41.27	48.41	43.25	60.71	51.59	14.29	**88.89**
5	75.16	69.57	**98.76**	44.10	76.40	75.16	75.78	96.89	79.50
6	55.90	41.49	68.56	6.99	76.86	86.90	**90.83**	27.95	82.53
7	82.86	0.00	**100.00**	96.19	97.14	**100.00**	**100.00**	**100.00**	98.10
8	76.32	**92.82**	62.68	81.10	88.04	56.94	77.99	77.99	91.15
9	59.62	0.00	70.77	66.15	88.08	62.69	68.85	77.89	**92.69**
10	93.56	35.64	96.04	25.25	94.80	99.75	99.26	94.80	**100.00**
11	51.07	**99.28**	86.87	89.50	98.33	89.02	94.27	90.22	98.57
12	61.83	34.99	60.64	89.26	88.87	41.55	81.11	**90.06**	88.07
13	98.60	95.25	**100.00**	93.96	99.89	99.25	99.89	**100.00**	**100.00**
OA(%)	74.92	62.10	81.55	76.18	87.86	78.27	81.26	83.20	**94.04**
AA(%)	71.76	51.96	79.75	70.27	84.34	76.76	81.59	78.39	**92.60**
κ(%)	72.12	57.88	79.53	73.44	86.51	75.92	79.32	81.35	**93.37**

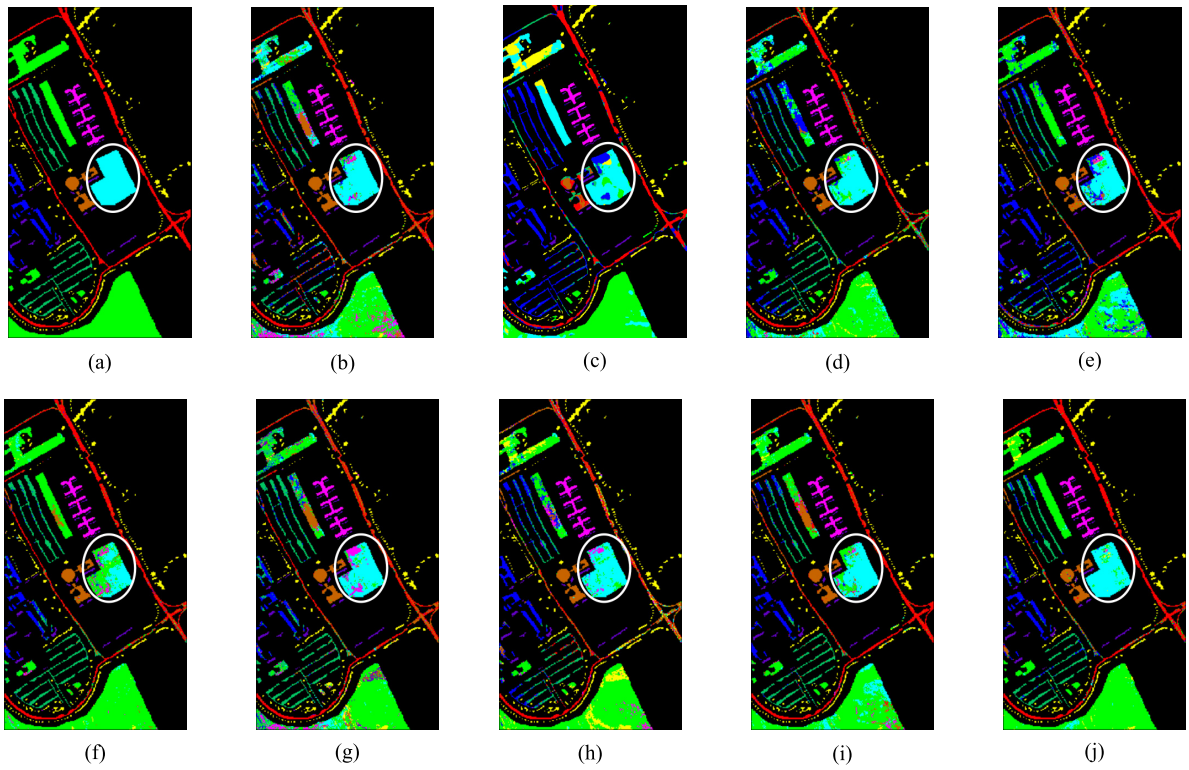

Fig. 5. Classification maps of different methods over the PU dataset. The ROIs are circled in white. (a) Ground truth. (b) EMP-SVM [10]. (c) RSSAN [30]. (d) RN-FSC [47]. (e) SS-CNN [41]. (f) AROC-DPNet [43]. (g) 3-DCAE [32]. (h) WGAN-GP [39]. (i) DMVL [40]. (j) XDCL.

task to learn the representations invariant under different band groups.

We reimplement the comparison methods by Pytorch. The parameters and settings of these methods are set to follow the suggestions of the original articles to ensure the best performance. Note that these methods are trained with the exact same labeled samples for a fair comparison.

The quantitative results over three datasets are presented in Tables II–IV, respectively. As can be seen, compared to the state-of-the-art methods, our method achieves the best performance over all the datasets in terms of OA, AA, and kappa values κ. In particular, although belonging to the same category, the XDCL outperforms the other unsupervised learning methods, which demonstrates the superiority of encoding the information shared between the spectral and spatial domains over compressing all the information of the HSI samples for learning powerful representations. Among them, DMVL gives better performance than 3-DCAE and WGAN-GP, which is because it captures more semantics by exploring the information invariant under different bands. However, it still does not recognize much of the nuisance information and learns weaker representations than the XDCL.

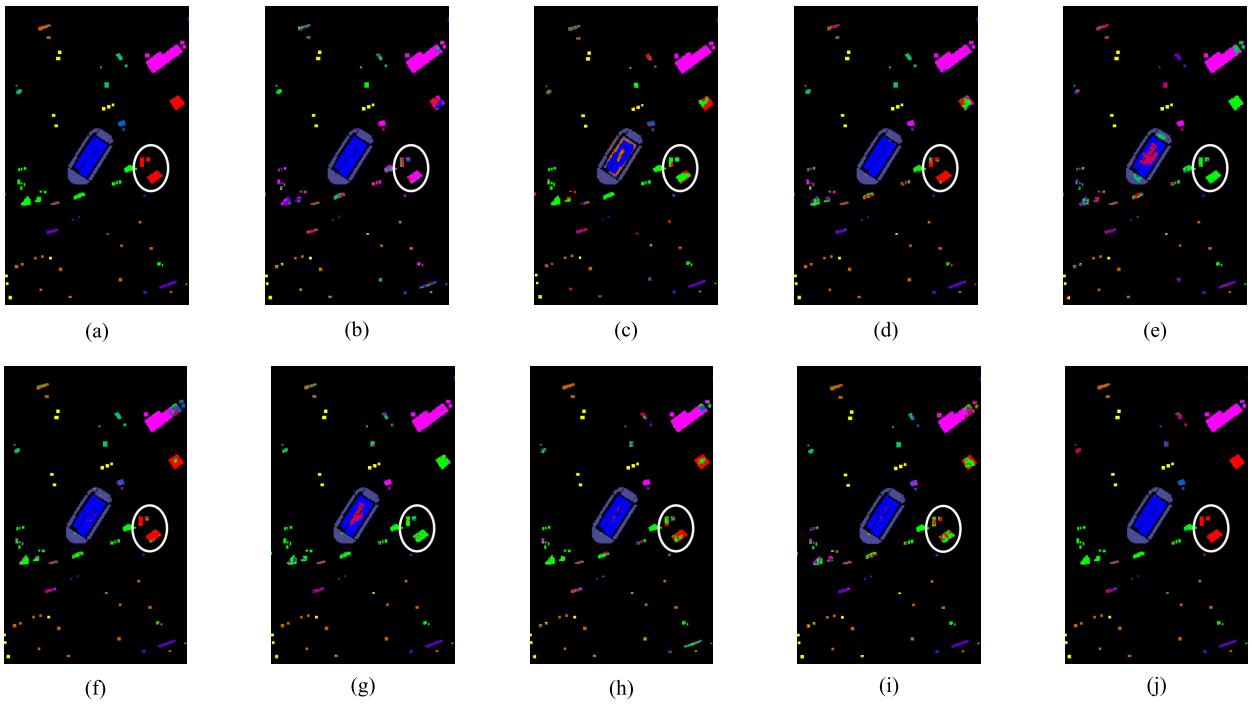

Fig. 6. Classification maps of different methods over the Houston dataset. The ROIs are circled in white. (a) Ground truth. (b) EMP-SVM [10]. (c) RSSAN [30]. (d) RN-FSC [47]. (e) SS-CNN [41]. (f) AROC-DPNet [43]. (g) 3-DCAE [32]. (h) WGAN-GP [39]. (i) DMVL [40]. (j) XDCL.

As a supervised learning method, RSSAN has the lowest accuracy over all the datasets, which shows that the lack of labeled data limits the capability of deep networks for feature extraction greatly. It is very likely to encounter the overfitting problem while training a deep network with very few labels. In addition, the SS learning method AROC-DPNet has the second best performance, which illustrates that selecting samples with high-confidence pseudolabels as new labeled data is an effective way to extend the dataset. With regard to class-specific accuracy, the XDCL yields the best or nearly the best results in most classes while achieving relatively low accuracy in very few categories, such as category 3 in the PU dataset. In contrast, RSSAN, RN-FSC, and SS-CNN provide higher accuracy in this class. However, they perform very poorly in many of the remaining classes, which also results in their poor overall performance. Such highly variable performance across different classes may be due to overfitting. In comparison, XDCL alleviates this problem effectively and gives a much more stable performance across classes.

In order to evaluate the classification results more clearly, the classification maps of different methods and the ground truth over the three datasets are shown in Figs. 5–7, respectively. For better visual effects, we only present a part of the Houston and KSC datasets. Moreover, one region of interest (ROI) of each image is marked by a white circle for a clearer comparison. Overall, the XDCL generates classification maps with the fewest classification errors on the three datasets. Taking the PU dataset as an example, as shown in the ROIs circled in white, the XDCL presents the smoothest and most accurate results, while the comparison methods show severe salt-and-pepper alike defects where many pixels are misclassified. Comparing the classification maps of different methods over the other two datasets shows similar results. It demonstrates that encoding the domain-invariant information could help us overcome the variability in spectral or spatial domains.

TABLE V
OVERALL CLASSIFICATION ACCURACY WITH DIFFERENT STRATEGIES FOR VISUAL SIGNAL CONSTRUCTION OVER THREE HSI DATASETS. THE BEST VALUE IS IN BOLD

Strategy	PU	Houston	KSC
Fixed band	81.48%	75.96%	85.41%
Average band	82.35%	76.64%	85.63%
PCA 1st comp	90.14%	84.31%	93.01%
Random selection	**91.27%**	**85.02%**	**94.04%**

E. Analysis of the Signal Construction

We use the center spectrum as the spectral signal and pick a band with the random selection strategy as the visual signal. The center spectrum containing spectral information of the object is the preferred choice. In this subsection, we mainly analyze the effect of our strategy to the visual signal construction from two perspectives: the band number and the band selection strategy.

In the first experiment, we vary the number of bands and evaluate the representation quality over the PU dataset. We present the result in Fig. 8. As can be seen, the OA decreases obviously as the number of bands increases. Using more bands mixes undesired spectral information with the visual signals. Such mixed information is shared between the two signals and provides a shortcut to solve the cross-domain

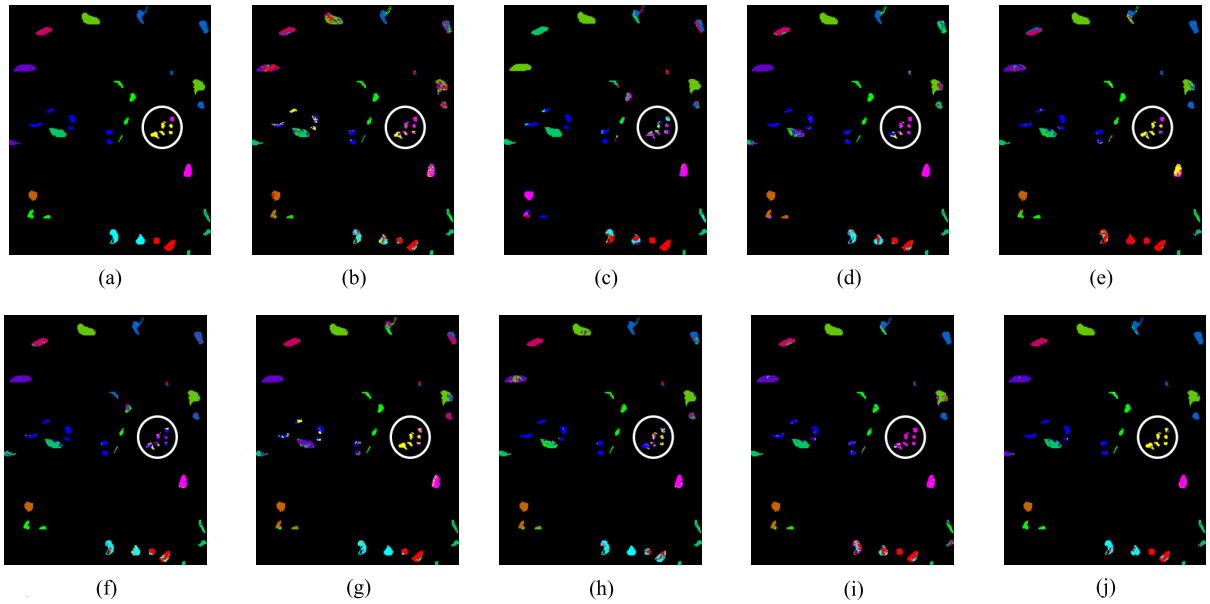

Fig. 7. Classification maps of different methods over the KSC dataset. The ROIs are circled in white. (a) Ground truth. (b) EMP-SVM [10]. (c) RSSAN [30]. (d) RN-FSC [47]. (e) SS-CNN [41]. (f) AROC-DPNet [43]. (g) 3-DCAE [32]. (h) WGAN-GP [39]. (i) DMVL [40]. (j) XDCL.

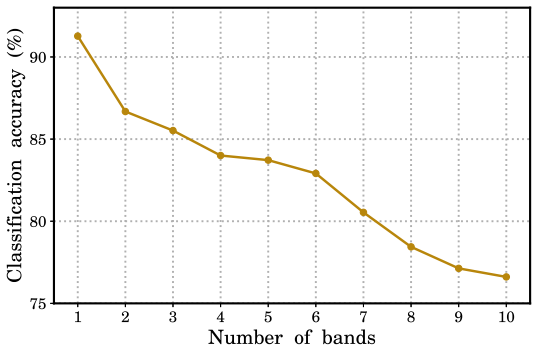

Fig. 8. Overall classification accuracy with visual signals including different number of bands over the PU dataset.

TABLE VI
OVERALL CLASSIFICATION ACCURACY WITH DIFFERENT MASKING STRATEGIES OVER THREE HSI DATASETS. THE BEST VALUE IS IN BOLD

Masking strategy	PU	Houston	KSC
No masking	88.28%	83.44%	92.72%
Grid-wise masking	89.03%	84.01%	93.19%
Point-wise masking	89.98%	84.46%	93.85%
Block-wise masking	**91.27%**	**85.02%**	**94.04%**

discrimination task. The base encoders would not struggle to find the high-level factors buried in the shared information, and the representation quality is, thus, lowered. The result demonstrates the superiority of using a single band to construct the visual signals.

In the second experiment, we study the effectiveness of our band selection strategy by comparing it with three other options. The first one uses a fixed band, e.g., the first band, as the signal. The second one constructs the signal by averaging all the bands. The last one applies PCA transformation to the sample and selects the first component, which is called "PCA 1st comp." The results over three datasets are shown in Table V. As can be seen, using a fixed band or the average band both incur significant performance drops, which is because insufficient information is included in the signals. Compared to them, selecting the first principal component obtains higher accuracy since it exploits more spatial information. In contrast, our strategy covering the maximum information and enhancing the data diversity gives the best performance over all the datasets, which verifies the effect of our strategy.

F. Analysis of the Masking Strategy

We design a blockwise masking strategy to randomly remove blocks of the signal, i.e., set part of it to zero, to reduce its redundancy. The masked signals are then used as the inputs of the base encoders. In addition to setting large blocks to zero, there are some alternatives for signal masking. Here, we present two other masking strategies. The first one randomly sets every point of a signal to zero with probability r, which works like the dropout operation in deep learning. We simply call this strategy "pointwise masking." This strategy randomly samples the masked points of the visual signal and the masked channels of the spectral signal with a ratio r. One important property of natural signals is the locality of point dependencies, where the neighboring points tend to be correlated. The CNN is able to make use of such a property and recover a missing point from nearby points easily. Compared to the pointwise masking, our strategy makes such extrapolation much harder, which pushes the base encoders to learn more high-level understanding rather than extracting low-level contents. The other strategy regularly masks n of every m point of the signals with a masking ratio $r = n/m$, which is referred to as "gridwise masking."

TABLE VII
TRAINING AND FEATURE EXTRACTION TIME OF DIFFERENT METHODS ON THE PU DATASET. THE BEST VALUE IS IN BOLD

Time	RSSAN [30]	RN-FSC [47]	SS-CNN [41]	AROC-DPNet [43]	3DCAE [32]	WGAN-GP [39]	DMVL [40]	XDCL
Training (min)	**0.06**	123.1	15.99	52.61	18.35	96.20	184.68	27.18
Feature extraction (s)	6.49	11.86	**1.43**	77.74	5.56	58.56	95.21	3.64

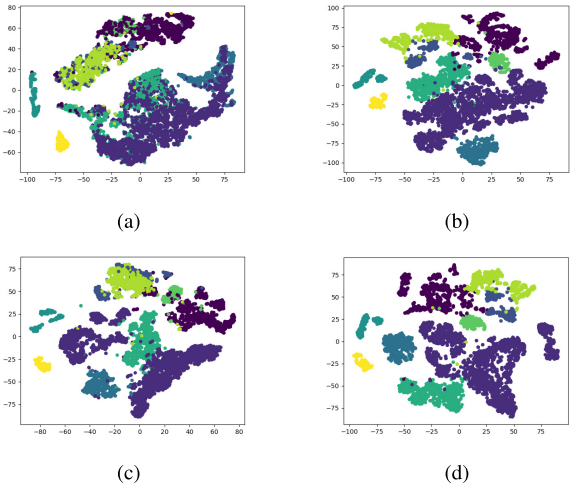

Fig. 9. PU dataset: feature visualization results of (a) raw spectral vectors, (b) representations of visual signals, (c) representations of spectral signals, and (d) concatenation of visual and spectral representations.

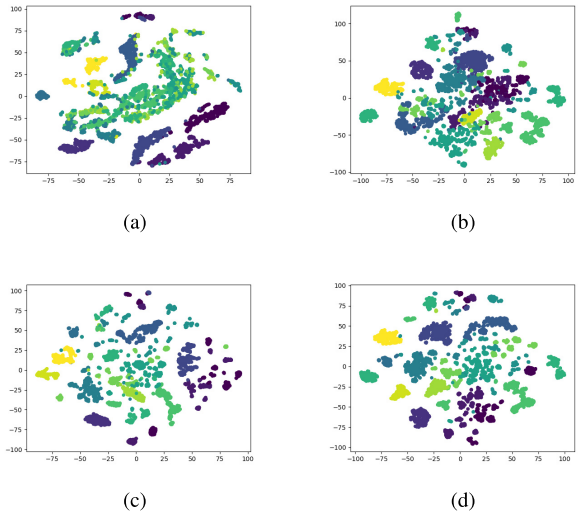

Fig. 10. Houston dataset: feature visualization results of (a) raw spectral vectors, (b) representations of visual signals, (c) representations of spectral signals, and (d) concatenation of visual and spectral representations.

In this set of experiments, the masking ratios of the blockwise and pointwise masking are set to 0.35. For the gridwise masking, n and m are set to 1 and 3, respectively. The results are shown in Table VI. As can be seen, there is an obvious performance drop when no mask is used, while masking the signals with different strategies can all lead to better results. Compared to gridwise masking, pointwise masking obtains higher accuracy, which validates the superiority of random sampling of masked points over regular sampling. Moreover, blockwise masking gives a better performance than pointwise masking, which verifies the advantage of our strategy over the others in reducing signal redundancy for learning strong representations.

G. Analysis of the Computational Complexity

In this subsection, we evaluate the computational complexity of the XDCL. We record the training and feature extraction time of different methods on the PU dataset. The experiments are conducted on the same computation platform, as stated in Section III. The results are presented in Table VII.

As can be seen, the supervised learning method, RSSAN, costs much less time than the others because it only uses very few labeled samples for the network training. However, the performance of supervised learning is very poor according to the previous experiments. Among all the other methods, XDCL gives the third best performance, yet it a significant amount of training time compared to RN-FSC and DMVL. Nevertheless, more training time is usually not a major problem for practical HSI classification since the training process is implemented offline. On the other hand, the feature extraction time is a more important metric for measuring the feasibility. For this, XDCL takes the second least time compared to the others since a very light network is used, and the size of the input sample to our model is relatively small. Taking both the classification accuracy and computational complexity into consideration, the XDCL is a very effective and efficient method for HSI classification with limited labeled samples.

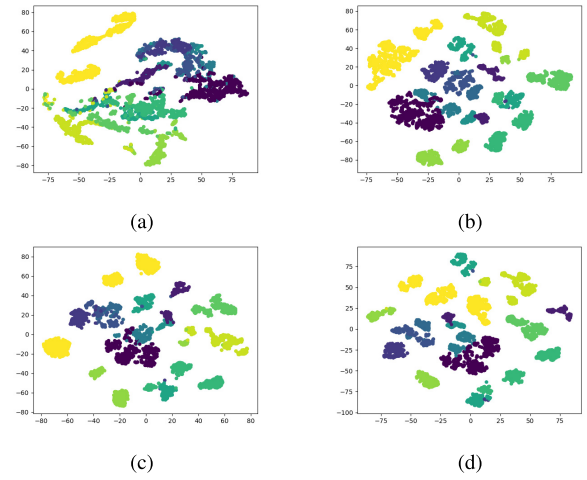

Fig. 11. KSC dataset: feature visualization results of (a) raw spectral vectors, (b) representations of visual signals, (c) representations of spectral signals, and (d) concatenation of visual and spectral representations.

H. Feature Visualization

In order to further evaluate the effectiveness of the XDCL, we use the t-distributed stochastic neighbor embedding (t-SNE) method [60] to visualize the learned representations of visual and spectral signals, and the concatenation of both, respectively. For comparison, we also visualize the raw spectral vectors. The results on three datasets are presented in Figs. 9–11, respectively. Taking the PU dataset as an example, the raw spectral vectors have quite poor separability, and the samples from different classes are heavily entangled with each other. In contrast, the separability of our visual and spectral representations is improved, which demonstrates the effectiveness of the XDCL in extracting useful semantic information. It is also observed that the samples belonging to different categories are further separated by concatenating the two representations.

V. Conclusion

In this work, we propose an unsupervised method called XDCL to learn HSI representations by contrasting spectral against spatial domains. One major obstacle of unsupervised learning is how to distinguish and extract useful information without labels. We show that the semantics is shared across different domains and propose to encode the shared information to discover valuable features. We carefully study how to construct ideal signals representing the two domains and present the effects of different strategies. We also note that removing parts of the signal further improves the representation quality. Our method outperforms the state-of-the-art methods on classification tasks with very few labeled samples. One disadvantage of the proposed method is that the training procedure with abundant unlabeled samples consumes a relatively long time, an issue that we will further investigate with lighter models and designing more efficient contrastive algorithms. This work opens up a new avenue for learning HSI representations suitable for classification. We hope that this will bring more inspiration to future work.

References

[1] W. Dong *et al.*, "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2337–2352, May 2016.

[2] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[3] M. B. Stuart, A. J. S. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, Jul. 2019.

[4] T. A. Carrino, A. P. Crósta, C. L. B. Toledo, and A. M. Silva, "Hyperspectral remote sensing applied to mineral exploration in southern Peru: A multiple data integration approach in the Chapi Chiara gold prospect," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 64, pp. 287–300, Feb. 2018.

[5] X. Tong, H. Xie, and Q. Weng, "Urban land cover classification with airborne hyperspectral data: What features to use?" *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 10, pp. 3998–4009, Oct. 2014.

[6] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[7] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, Mar. 2005.

[8] G. Mercier and M. Lennon, "Support vector machines for hyperspectral image classification with spectral-based kernels," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 1, Jul. 2003, pp. 288–290.

[9] A. Villa, J. Chanussot, C. Jutten, J. A. Benediktsson, and S. Moussaoui, "On the use of ICA for hyperspectral image analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 4, Jul. 2009, pp. IV-97–IV-100.

[10] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *Int. J. Remote Sens.*, vol. 31, no. 22, pp. 5975–5991, Dec. 2010.

[11] S. Jia, L. Shen, and Q. Li, "Gabor feature-based collaborative representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1118–1129, Feb. 2015.

[12] L. He, J. Li, A. Plaza, and Y. Li, "Discriminative low-rank Gabor filtering for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1381–1395, Mar. 2017.

[13] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[14] C.-I. Chang and H. Ren, "An experiment-based quantitative and comparative analysis of target detection and image classification algorithms for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 2, pp. 1044–1063, Mar. 2000.

[15] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1139–1155, Apr. 2009.

[16] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based K-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[17] G. Chen and S.-E. Qian, "Dimensionality reduction of hyperspectral imagery using improved locally linear embedding," *J. Appl. Remote Sens.*, vol. 1, no. 1, 2007, Art. no. 013509.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[19] T. Zeng, H. K.-H. So, and E. Y. Lam, "Computational image speckle suppression using block matching and machine learning," *Appl. Opt.*, vol. 58, no. 7, p. B39, Mar. 2019.

[20] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[21] N. Meng, E. Y. Lam, K. K. Tsia, and H. K.-H. So, "Large-scale multi-class image-based cell classification with deep learning," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 5, pp. 2091–2098, Sep. 2019.

[22] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[23] Y. Zhu, C. H. Yeung, and E. Y. Lam, "Microplastic pollution monitoring with holographic classification and deep learning," *J. Phys., Photon.*, vol. 3, no. 2, Apr. 2021, Art. no. 024013.

[24] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, Jul. 2015, Art. no. 258619.

[25] X. Li, M. Ding, and A. Pižurica, "Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2615–2629, Apr. 2020.

[26] B. Zhang, L. Zhao, and X. Zhang, "Three-dimensional convolutional neural network model for tree species classification using airborne hyperspectral images," *Remote Sens. Environ.*, vol. 247, Sep. 2020, Art. no. 111938.

[27] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[28] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jan. 2015.

[29] C. Zhang, G. Li, and S. Du, "Multi-scale dense networks for hyperspectral remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9201–9222, Aug. 2019.

[30] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 449–462, Jan. 2021.

[31] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[32] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[33] P. Guan and E. Y. Lam, "Multistage dual-attention guided fusion network for hyperspectral pansharpening," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2472–2481.

[35] N. Meng, H. K.-H. So, X. Sun, and E. Y. Lam, "High-dimensional dense residual convolutional neural network for light field reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 43, no. 3, pp. 873–886, Mar. 2021.

[36] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.

[37] 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.

[38] M. Zhang, M. Gong, Y. Mao, J. Li, and Y. Wu, "Unsupervised feature extraction in hyperspectral images based on Wasserstein generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2669–2688, May 2019.

[39] Q. Sun and S. Bourennane, "Hyperspectral image classification with unsupervised feature extraction," *Remote Sens. Lett.*, vol. 11, no. 5, pp. 475–484, Feb. 2020.

[40] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[41] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, Sep. 2017.

[42] Y. Cai, Z. Zhang, Z. Cai, X. Liu, and X. Jiang, "Hypergraph-structured autoencoder for unsupervised and semisupervised classification of hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[43] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 164–178, Mar. 2020.

[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1199–1208.

[45] M. Ruswurm, S. Wang, M. Korner, and D. Lobell, "Meta-learning for few-shot land cover classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 200–201.

[46] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[47] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sens.*, vol. 12, no. 6, p. 923, Mar. 2020.

[48] Y. Zhu, C. Hang Yeung, and E. Y. Lam, "Digital holographic imaging and classification of microplastics using deep transfer learning," *Appl. Opt.*, vol. 60, no. 4, p. A38, Feb. 2021.

[49] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2020, pp. 1597–1607.

[50] H. Zhang, J. Y. Koh, J. Baldridge, H. Lee, and Y. Yang, "Cross-modal contrastive learning for text-to-image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 833–842.

[51] P. Morgado, N. Vasconcelos, and I. Misra, "Audio-visual instance discrimination with cross-modal agreement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12475–12486.

[52] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 776–794.

[53] C. Zhuang, T. She, A. Andonian, M. Sobol Mark, and D. Yamins, "Unsupervised learning from video with deep neural embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9563–9572.

[54] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 9, May 2010, pp. 297–304.

[55] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 15535–15545.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015, pp. 1–15.

[57] P. Zhang and E. Y. Lam, "From local to global: Efficient dual attention mechanism for single image super-resolution," *IEEE Access*, vol. 9, pp. 114957–114964, 2021.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[59] Z. Ren, H. K.-H. So, and E. Y. Lam, "Fringe pattern improvement and super-resolution using deep learning in digital holography," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 6179–6186, Nov. 2019.

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

Peiyan Guan (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from Nanjing University, Nanjing, China, in 2018. He is pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong.

His research interests include hyperspectral image analysis, pattern recognition, and machine learning.

Edmund Y. Lam (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1995, 1996, and 2000, respectively.

From 2010 to 2011, he was a Visiting Associate Professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. He is a Professor of electrical and electronic engineering with The University of Hong Kong, Hong Kong, where he also serves as the Computer Engineering Program Director. His main research is in computational imaging.

Dr. Lam is also a Fellow of the Optical Society of America (OSA), the Society of Photo-Optical Instrumentation Engineers (SPIE), the Society for Imaging Science and Technology (IS&T), and the Hong Kong Institution of Engineers (HKIE). He was a recipient of the IBM Faculty Award.