Generative Adversarial Minority Oversampling for Spectral-Spatial Hyperspectral Image Classification

Swalpa Kumar Roy, *Student Member, IEEE,* Juan M. Haut, *Senior Member, IEEE,* Mercedes E. Paoletti, *Graduated Member, IEEE,* Shiv Ram Dubey, *Member, IEEE,* Antonio Plaza, *Fellow, IEEE*

***Abstract*—Recently, convolutional Neural Networks (CNNs) have exhibited commendable performance for hyperspectral image (HSI) classification. Generally, an important number of samples are needed for each class to properly train CNNs. However, existing HSI datasets suffer from a high class imbalance, where many classes do not have enough samples to characterize the spectral information. The performance of existing CNN models is biased towards the majority classes having more samples for the training. This paper addresses this issue of imbalanced data in HSI classification. A new `3D-HyperGAMO` model is proposed, which uses the generative adversarial minority oversampling. The proposed `3D-HyperGAMO` generates more samples for minority classes (automatically) at training time, using the existing samples of that class. The samples are generated in the form of a 3D hyperspectral patch. A classifier other than the generator and discriminator is used in the `3D-HyperGAMO` model which is trained using both original samples and generated samples to find the classes of new generated samples where actually it belongs. The generated data are combined class-wise with original training dataset to learn the parameters of the class network. Finally, the trained 3D classifier network validates the performance of the model using the test set. Four benchmark HSI datasets, namely Indian Pines (IP), Kennedy Space Center (KSC), University of Pavia (UP), and Botswana (BW), have been considered in our experiments. The proposed model shows outstanding data generation ability during training, which significantly improves classification performance over the considered datasets. The source code will be available publicly at https://github.com/mhaut/3D-HyperGAMO.**

***Index Terms*—Deep Learning, convolutional neural networks, spectral-spatial hyperspectral image classification.**

I. Introduction

HYPERSPECTRAL images (HSI) capture the spectral information reflected by materials by gathering different wavelengths along the electromagnetic spectrum in hundreds of narrow, contiguous bands. The resulting data cube provides valuable information about the observed materials, where each pixel represents the spectral signature of each imaged object. HSI analysis has been successfully applied in several applications related to Earth observation and remote sensing such as vegetation modeling, urbanisation analysis, crop analysis, and many more [1]. Several attempts have been conducted for HSI classification using classical pattern recognition and machine learning methods, such as kernel-based methods [2], composite kernels [3], dictionary-based sparse representation [4], multiple feature learning [5], fusion of correlation coefficient and joint sparse representation [6], etc. However, these approaches usually depend on handcrafted features, which are difficult to obtain and depend on expert knowledge.

In recent years, traditional feature representation and learning methods have been superseded by convolutional neural networks (CNNs) [7]. The CNN is a hierarchical feature learning and classification technique whose architecture is composed by a stack of different layers including convolution and activation functions [8]. In this sense, the bottom layers provide simple and low-abstract features from the input data, which are further sophisticated as they are processed by deeper layers, until the top layers obtain very representative and abstract features. In this way, the CNN extracts the most relevant features automatically from the training data, exhibiting outstanding results in a wide range of computer vision applications [9] and text analysis [10], among others. In particular, CNNs have shown a great performance in remote sensing HSI classification problems. Many CNN-based models are currently available in the HSI classification literature. For instance, some papers combine CNN models with handcrafted features, such as He *et al.* [11] who consider covariance matrices and CNN models. However, the vast majority of methods consider features automatically extracted by CNN models. In this context, an optimized CNN is proposed by Yu *et al.* [12], which applies three 1×1 convolutional layers with average pooling layer and larger dropout rates over $5 \times 5 \times B$ input HSI patches (here, B denotes the number of bands), reducing the spectral dimension while keeping constant the spatial dimension as the HSI feature volumes go deep through the model. Also, Paoletti *et al.* [13] have developed CNNs with three convolutional layers considering different input spatial patches and kernel sizes, by expanding the spectral domain and then reducing it to extract the most relevant spectral information while combining it with the contextual one. Makantasis *et al.* [14] have implemented a CNN2D for HSI classification, introducing randomized principal component analysis (R-PCA) [15] to extract the first 10 to 30 principal components in order to reduce the computational cost

This work has been supported by Junta de Extremadura (Decreto 14/2018, de 6 de febrero, por el que se establecen las bases reguladoras de las ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR18060) (*Corresponding author: Juan M. Haut*)

S.K. Roy is with Computer Science and Engineering Department at Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal-735102, India (e-mail: swalpa@cse.jgec.ac.in).

J.M. Haut, M. E. Paoletti and A. Plaza are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain (e-mail: juanmariohaut@unex.es; mpaoletti@unex.es; aplaza@unex.es).

S.R. Dubey is with Computer Vision Group, Indian Institute of Information Technology, Sri City, Chittoor, Andhra Pradesh-517646, India (e-mail: srdubey@iiits.in).

This paper is published by IEEE Transactions on Geoscience and Remote Sensing (IEEE-TGRS). The final paper is available at: https://ieeexplore.ieee.org/document/9347550.

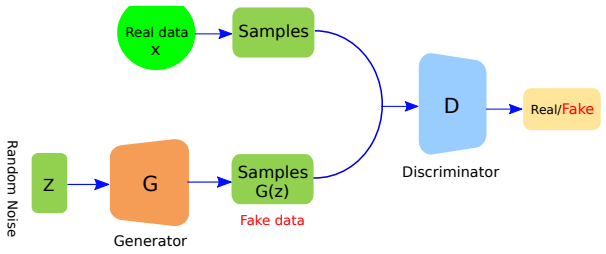


Fig. 1. Generative adversarial network (GAN) framework using generator and discriminator networks.

of training and prediction stages. Li *et al.* [16] have exploited the learning of the pixel-pair features of deep CNN models to increase their discriminative ability by combining the HSI patch center pixel with each of its neighboring pixels, applying at the end a voting strategy.

Also, more complex strategies have been combined with CNN-based models in order to enhance their FE ability. For instance, Haut *et al.* [17] and Cao *et al.* [18] apply active learning to obtain the most discriminative features to enhance the classification performance. Currently, residual learning [19] has gained a lot of popularity, since a wide variety of models and architectures have been succesfully applied within HSI data classification problems. For instance, Kang *et al.* [20] have developed a dual-path CNN for HSI classification by combining different levels of features with residual learning. Also, Song *et al.* [21] have employed deep feature fusion with residual connections for the same purpose. Cheng *et al.* have also explored different kinds of features for HSI classification [22]. Hamida *et al.* [23] have employed the CNN3D for HSI classification, which has been further extended by He *et al.* to multi-scale CNN3D [24]. Zhong *et al.* have used the 3D residual network, developing the spectral-spatial residual network (SSRN) [25]. Paoletti *et al.* [26] have proposed a new residual pyramid bottleneck to keep the computational loads balanced along the model, elongating the spectral domain while reducing the spatial shape of obtained features. Also Paoletti *et al.* [27] extended the reutilization of previous features by developing a densely connected residual model. Moreover, several works such as Haut *et al.* [28], Fang *et al.* [29], Xu *et al.* [30], Wu *et al.* [31] and Gao *et al.* [32] combine residual connections with attention mechanism to enhance the FE process, highlighting the most useful features for HSI data classification and attenuating the less relevant ones. Recently, Roy *et al.* have proposed the HybridSN to utilize both spectral and spatial features by sequentially combining 3D and 2D convolutions [33] and lightweight spectral-spatial squeeze-and-excitation residual bag-of-features learning to reduce the trainable parameters for HSI classification [34]. Moreover, Zheng *et al.* have developed a fast patch-free global learning CNN for HSI classification [35]. Liu *et al.* have introduced content-guided CNNs for HSI classification [36]. Although several methods using CNN have been introduced for HSI classification, these methods are unable to tackle the fundamental problem of class imbalance which is often present in HSI datasets.

In addition to convolution-based models and residual connections, the deep learning field has been revolutionized by the development of a new type of deep architecture, known as generative adversarial networks (GANs). GANs were originally proposed by GoodFellow *et al.* in 2014 [37] in order to facilitate the generation of new samples in the given probability distribution. The GAN uses two networks, namely generator and discriminator, as depicted in Fig. 1. The generator network provides new samples (i.e., fake samples) from a random noise vector, while the discriminator network recognises the real and fake samples. The whole GAN is trained in such a way that the generator tries to fool the discriminator by generating more realistic samples, and the discriminator tries not to get fooled by generator by classifying the generated samples as fake ones. Thus, a kind of adversarial learning (like a mini-max game) is performed to train both the generator and discriminator networks. Very recently, generative adversarial learning has been applied to remotely sensed HSI data to perform several tasks, such as data semantic segmentation [38], super-resolution [39], and anomaly detection [40], [41], among others.

In last three years, a significant number of works have been reported for HSI classification using adversarial training. For instance, Zhang *et al.* have introduced a semisupervised HSI classification framework using a 1D GAN for HSI (HSGAN) [42]. The HSGAN uses the discriminator features for the classification of HSI data. Zhu *et al.* have used the GAN to generate new samples which are used for training the network for HSI classification. The spectral features are generated using a custom 1-D GAN in [43], which is used by another CNN for HSI classification with the help of majority voting. More recently, in 2019 Feng *et al.* have developed a multiclass spatial-spectral GAN (MSGAN) [44] by employing two generators for spatial and spectral information along with the adversarial objectives for multiple classes. A Caps-TripleGAN model has been investigated by Wang *et al.* [45] by generating the samples using a 1-D structure triple generative adversarial network (TripleGAN) and classifying the HSI data using the capsule network (CapsNet) [46]. Xue *et al.* have proposed a GAN-based HSI classification approach by utilizing a deep 3D-CNN based generator network and another 3D deep residual network based discriminator network [47]. The GAN has also been explored with conditional random fields (CRFs) for HSI classification [48]. The dense CRFs refine the generated HSI classification map. A DropBlock structure based generative adversarial network (DBGAN) is investigated by Yin *et al.* [49] to generalize the HSI classification by randomly dropping the blocks. An adversarial training driven hallucination architecture is proposed by Pande *et al.* [50] to learn the features corresponding to the missing modalities for remote sensing. Wang *et al.* have performed adversarial training to generate spectral samples [51], then used for HSI classification. The CapsNet and convolutional long short-term memory (ConvLSTM) based discriminator network has been used by Wang *et al.* [52] to learn high-level contextual features for HSI classification.

Although the aforementioned GAN-based methods use adversarial training for HSI classification, they do not consider the minority classes. Generating the samples for each class or

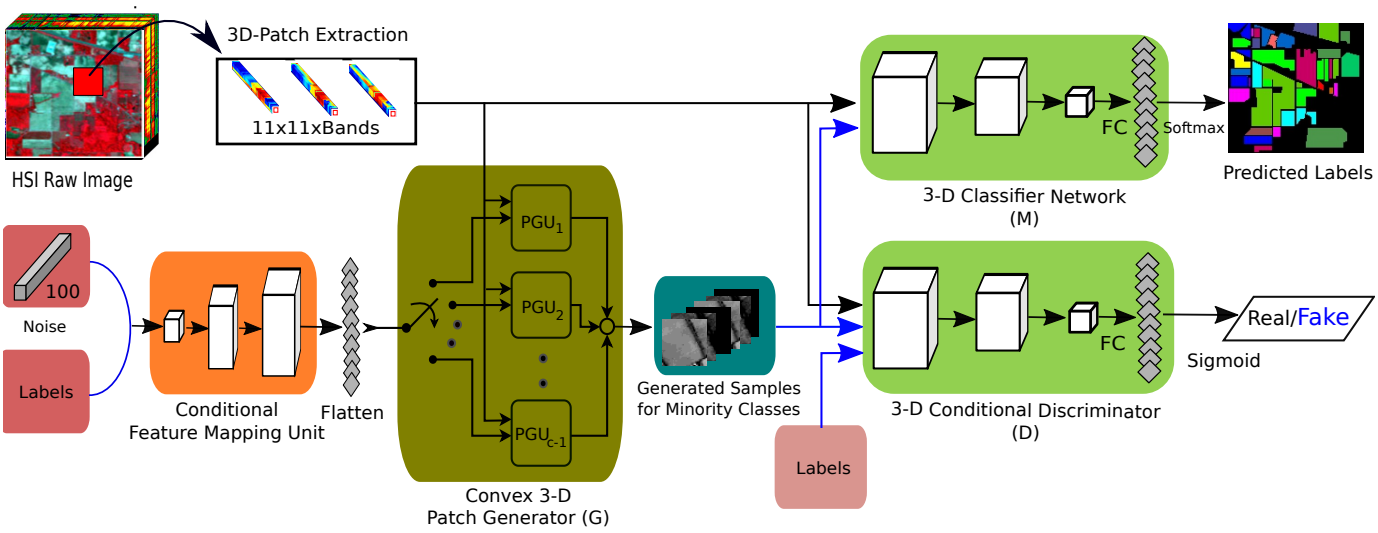


Fig. 2. Graphical overview of the proposed 3D-HyperGAMO architecture for generative adversarial minority oversampling when dealing with imbalanced HSI classification problems. It comprises a 3D HSI patch extractor, a conditional feature mapping unit, a 3D HSI patch generator to generate new data patches, a 3D HSI conditional discriminator and a 3D HSI classification network.

spectral band does not solve the class imbalance problem in HSI datasets. Recently, Mullick *et al.* have introduced generative adversarial minority oversampling (GAMO) to generate the samples using a 1D generator and a 1D discriminator [53]. Motivated by the problem of class imbalance in HSI datasets and the success of GAMO model, in this paper we introduce a new `3D-HyperGAMO` model to deal with the class imbalance problem in HSI classification. The main contributions of the paper can be summarized as follows:

1) We propose a new `3D-HyperGAMO` model to tackle the long-standing and most challenging problem of class imbalance for the first time in HSI data literature.
2) The proposed model generates a sufficient number of samples for minority classes by considering an oversampling strategy with the help of the convex 3D patch generator, which is guided by another discriminator network following an adversarial learning manner.
3) The proposed method uses a conditional feature mapping unit and a convex 3D patch generator (G) to generate the HSI patches within the boundary of probability distribution of minority classes, and a 3D discriminator (D) to distinguish between real and fake HSI samples.
4) The proposed model also uses a 3D classifier network (M) to classify the 3D HSI patches into the corresponding classes.
5) Rigorous experiments are conducted over four HSI benchmark datasets, reaching improved classification performance with very low amounts of training data.

The remainder of the paper is organised as follows. Section II introduces the proposed `3D-HyperGAMO` model. Our experimental results are presented in Section III, along with a detailed discussion. Section IV concludes the paper with some remarks and hints at plausible future research lines.

II. Proposed 3D-HyperGAMO Architecture

The HSI classification task suffers from the presence of imbalanced data for different classes. Generally, some percentage of the total data for each class is taken for the training of the model. Most of the available HSI datasets are highly imbalanced, and training sets obtained by randomly taken samples is also imbalanced. This leads to bias in the training of any CNN model towards the majority classes. In other words, the minority classes are not properly learned by the network due to the weak representation caused by the insufficient number of samples.

Here, we propose a `3D-HyperGAMO` model inspired by GAMO model [53]. The `3D-HyperGAMO` is proposed to oversample 3D HSI patches from the minority classes using a generative adversarial approach. Basically, our `3D-HyperGAMO` generates the new samples in the distribution of existing class-specific samples for minority classes from noise. The proposed `3D-HyperGAMO` model and all the attributes of the framework are illustrated in Fig. 2. The proposed framework consists of the following main components: 1) 3D-patch extractor (which takes full advantage of all the spectral information contained in the HSI data cube channels), 2) a conditional feature mapping unit to generate the intermediate features from noise, 3) a convex 3D hyperspectral patch generator unit to generate the new samples of minority classes, 4) a 3D hyperspectral conditional discriminator unit for the recognition of real versus fake 3D patches, and 5) a 3D CNN classification network unit to perform the classification of the original and the generated 3D HSI patches into categories.

A. 3D-Patch Extractor

A HSI can be represented as a 3D tensor with W, H, and B being the width, height and number of spectral bands, and defined as $\mathbf{X_{orig}} \in \mathcal{R}^{W \times H \times B}$. The HSI classification problem is generally performed by classifying all the pixel points into

L_c land-cover classes, denoted by $Y = (y_1, y_2, \dots y_{L_c})$. The pixel $\mathbf{x}_{i,j} \in \mathbf{X_{orig}}$ where $i = 1, \dots W$ and $j = 1, \dots H$ is defined as a spectral vector $\mathbf{x}_{i,j} = [x_{i,j,1}, \dots x_{i,j,B}] \in \mathcal{R}^B$ which represents the spectral signature of the captured surface materials. The 3D HSI patches ($\mathbf{h}_i \in \mathcal{R}^{S \times S \times B}$) of dimension $S \times S \times B$ are extracted, where $S \times S$ is the spatial dimension and B is the number of channels. Basically, a sliding window approach with stride s is used over the spatial domain across each channel to generate the 3D HSI patches, which are treated as the input to a convex 3D patch generator unit and a 3D classifier network. In this sense, each extracted HSI patch takes full advantage of all the spectral information contained into HSI channels. To fulfill the desired goal of minority data oversampling using a deep learning-based framework which utilizes both the spectral-spatial information effectively, we stack all the patches into $\mathbf{X} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_i, ..., \mathbf{h}_k\}$ where $\mathbf{h}_i \in \mathcal{R}^{S \times S \times B}$ and k is the number of patches. In our experiments, a spatial window of size $S \times S \times B$ is defined (set empirically to $11 \times 11 \times Bands$ in the four considered data sets).

B. Conditional Feature Mapping Unit

The conditional feature mapping unit is used to generate the intermediate features (I_f) for the minority classes from a random vector (z). Consider $r^z \in \mathbb{R}_{\geq 0}$ as a random vector having z non-negative real numbers and $l \in \{0, 1\}$ as a hot encoding binary vector having c values to represent the label for class number, where c is the number of land-cover categories in the datasets. Only one value of l is one (i.e., the one corresponding to the class for which we want to generate the new samples) and other values are zero. The aim of this unit is to map the random vector (r) to the intermediate features conditioned on the label vector (l). In this unit, first the random vector (r) and then the label vector (l) are concatenated to form a resultant vector having $z + c$ values. Then, a dense layer is used to generate the features, with length $256 * 7 * 7 = 12544$, followed by a LeakyReLU activation function and Batch normalization using momentum of 0.9. It is reshaped into a $256 \times 7 \times 7$ 3D volume and then successively followed by a transpose of Conv2D layer to generate features of dimension $128 \times 5 \times 5$, with LeakyReLU activation function and batch normalization. The setting is used to generate the output of shape $1 \times 5 \times 5$ from the final transpose of the Conv2D layer, but with linear activation. Thus, the final output of this unit is flattened in the form of an intermediate conditional feature vector (I_f) of length 784. Note that the output of the conditional feature mapping unit is connected to one of the convex 3D patch generator units based on the label (l) information. The detailed structure and the parameters of conditional flatten feature mapping unit are shown in Table I.

C. Convex 3D Patch Generator Unit (PGU)

The convex 3D Patch Generator Unit (PGU) is proposed to oversample the data points from the minority classes as illustrated in Fig. 3. The class having the highest number of samples in the training set is considered as the majority class,

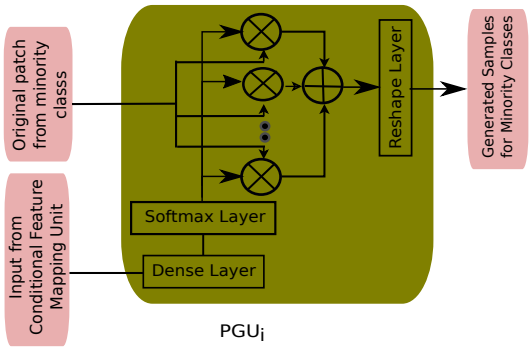


Fig. 3. The internal design architecture of convex patch generation unit.

whereas the remaining classes are considered as the minority classes. Thus, we use $c - 1$ PGU units, one for each minority class. Note that c is the total number of classes in the given dataset. The $PGU_i|_{i \in [1, c-1]}$ refers to the PGU unit for the i^{th} minority class. The PGU_i unit generates λ_i^g samples, such that:

$$\lambda_i^g = \lambda_m - \lambda_i \tag{1}$$

for $\forall i \in [1, c-1]$, where λ_m is the number of training samples in the majority class and λ_i is the number of training samples in the i^{th} minority class.

Any PGU unit takes two inputs, including the output of the conditional feature mapping unit (I_f having dimension: $784D$) and the 3D-patches of HSI data (dimension: $\lambda_i \times S \times S \times B$) where λ_i is the number of training patches from the i^{th} class. At a time only one PGU unit is active. The PGU_i will be used when the 3D-Patch is from the i^{th} minority class and the i^{th} binary bit (only) of label's hot encoding vector (l) is set to one. A dense layer is used to transform the intermediate feature (I_f) into a feature of length λ_i, followed by the $softmax$ activation function. This feature vector is repeated $\eta = S \times S \times B$ times to generate the class specific random feature matrix (I_m) of dimension $\eta \times \lambda$. The input 3D-patches of the HSI data (dimension: $\lambda_i \times S \times S \times B$) are also reshaped into a matrix (P_m) of dimension $\lambda \times \eta$. Finally, the class-specific feature matrix (F_m) of dimension $\eta \times \lambda$ is generated as follows:

$$F_m = I_m \cdot (F_m)^T \tag{2}$$

where $(\cdot)$ is the dot product and $(F_m)^T$ is the transpose of matrix F_m. A flattened vector (F_v) of dimension (η) is computed by taking the column-wise sum for each row of matrix F_m. The vector F_v is reshaped in order to generate the new 3D-patch (G_{3dp}) of dimension $S \times S \times B$ for the i^{th} class. Note that we refer to the generator network (G) as the combination of the conditional feature mapping unit and the 3D patch generator unit, as depicted in Fig. 2.

D. 3D Conditional Discriminator Unit

We use a 3D-Hyperspectral Conditional Discriminator (D) unit to differentiate between the original 3D-HSI patch and the generated 3D-HSI patch using the generator network (G). This network tries to classify the original patch into a "real class" and the generated patch into a "fake class."

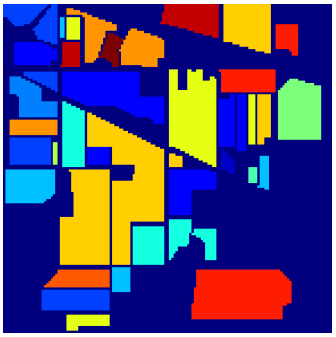
TABLE I
DETAILED STRUCTURE AND PARAMETERS OF 3D-HYPERGAMO ARCHITECTURE.

Conditional Flatten Feature Mapping				
Layer ID	**Kernel/Neurons**	**Stride**	**BatchNorm**	**Act. function**
Dense_1	256*7*7	-	Yes	LeakyReLU
Reshape_1	$256 \times 7 \times 7$	-	-	-
Conv2DTranspose_1	$128 \times 5 \times 5$	1	Yes	LeakyReLU
Conv2DTranspose_2	$64 \times 5 \times 5$	2	Yes	LeakyReLU
Conv2DTranspose_3	$1 \times 5 \times 5$	2	Yes	Linear
3D Conditional Discriminator				
Layer ID	**Kernel/Neurons**	**Stride**	**BatchNorm**	**Act. function**
Conv3D_1	$8 \times 1 \times 1 \times 7$	1	No	ReLU
Conv3D_2	$16 \times 3 \times 3 \times 5$	1	No	ReLU
Conv3D_3	$32 \times 5 \times 5 \times 7$	1	No	ReLU
ReshapeTo2D_1	-	-	-	-
Conv2D_1	$64 \times 3 \times 3$	1	No	ReLU
AveragePool_1	-	-	-	-
Dropout_1	0.5	-	-	-
Dense_1	1	-	-	Sigmoid
3D Classifier Network				
Layer ID	**Kernel/Neurons**	**Stride**	**BatchNorm**	**Act. function**
Conv3D_1	$8 \times 1 \times 1 \times 7$	1	No	ReLU
Conv3D_2	$16 \times 3 \times 3 \times 5$	1	No	ReLU
Conv3D_3	$32 \times 5 \times 5 \times 7$	1	No	ReLU
ReshapeTo2D_1	-	-	-	-
Conv2D_1	$64 \times 3 \times 3$	1	No	ReLU
AveragePool_1	-	-	-	-
Dropout_1	0.5	-	-	-
Dense_1	$n\_classes$	-	-	Softmax

This network consists of the following layers: 1) an input layer with original/generated 3D-HSI patch, 2) three 3D-Convolution layers with 8 filters having kernel sizes of (1, 1, 7), followed by 16 filters having kernel sizes of (3, 3, 5), followed by 32 filters having kernel sizes of (5, 5, 7), 3) a 3D activation that is reshaped into a 2D activation followed by a 2D convolution layer with 64 filters having kernel sizes of (3, 3), and 4) an average pooling layer followed by a dropout layer having 0.5 dropout factor leading to a final dense layer with a sigmoid activation function. The final dense layer produces the probability of being classified as a real patch. More information on the 3D Discriminator network is given in Table I.

E. 3D Classifier Network

The 3D Classifier Network (M) plays the main role throughout the 3D-HyperGAMO network, that needs to be trained for the HSI patch classification so that it can easily categorise the newly generated real samples as validated by the discriminator. During training of the classification network M, the original training samples of each class (as well as the generated valid samples from the PGU of the minority classes are used). During the testing stage, the test samples are needed to evaluate the performance of the classifier network. Thus, the Generator (G) and Discriminator (D) networks are not used for testing. However, the use of Generator (G) and Discriminator (D) networks during training increases the learning capacity of Classifier (M) network by including more samples for the minority classes, generated within the boundary of that class. The 3D network architecture of classifier network is the same as the 3D discriminator network (up to the dropout layer). The difference is that, after that layer, the classifier network uses a dense layer having c outputs followed by a softmax classifier layer. Basically, the final dense layer produces the class scores for each class, which are used to compute the softmax (cross-entropy) loss. A detailed summary is provided in Table I for the 3D-HSI Classifier network.



Color	Land cover type	Samples
	Background	10776
	Alfalfa	46
	Corn notill	1428
	Corn min	830
	Corn	237
	Grass/Pasture	483
	Grass/Trees	730
	Grass/pasture-mowed	28
	Hay windrowed	478
	Oats	20
	Soybeans notill	972
	Soybeans min	2455
	Soybean clean	593
	Wheat	205
	Woods	1265
	Bldg Grass Tree Drives	386
	Stone steel towers	93
	Total samples	21025

Fig. 4. The ground truth, type associated with the land-cover classes and the number of available samples in the Indian Pines (IP) dataset.

F. The Objective Function

The proposed method has three major networks: 3D-Patch Generator (G), 3D-Patch Discriminator (D), and 3D-Patch Classifier (M). The training is performed as a three-player mini-max game which is formulated as:

$$\min_{G} \max_{M} \max_{D} O(G, M, D) = \sum_{i \in c} O_i, \tag{3}$$

where, $O_i = (O_{i1} + O_{i2} + O_{i3} + O_{i4} + O_{i5} + O_{i6})$,

$$O_{i1} = P_i \mathbb{E}_{\mathbf{h} \sim p_i^d}[\log M_i(\mathbf{h})],$$

$$O_{i2} = \sum_{j \in c \setminus \{i\}} P_j \mathbb{E}_{\mathbf{h} \sim p_j^d}[\log(1 - M_i(\mathbf{h}))],$$

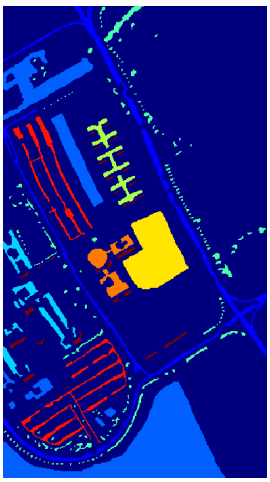
$$O_{i3} = (P_c - P_i) \mathbb{E}_{G(\mathbf{z}|i) \sim p_i^g}[\log M_i(G(\mathbf{z}|i))],$$

$$O_{i4} = \sum_{j \in c \setminus \{i\}} (P_c - P_j) \mathbb{E}_{G(\mathbf{z}|j) \sim p_j^g}[\log(1 - M_i(G(\mathbf{z}|j)))],$$

$$O_{i5} = P_i \mathbb{E}_{\mathbf{h} \sim p_i^d}[\log D(\mathbf{h}|i)],$$

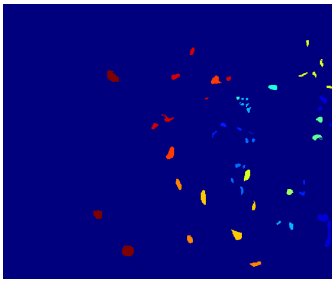
$$O_{i6} = (P_c - P_i) \mathbb{E}_{G(\mathbf{z}|i) \sim p_i^g}[\log(1 - D(G(\mathbf{z}|i)|i))],$$

while c is the number of classes, P_i is the prior probability of the i^{th} class, p_i^g and p_i^d, respectively denote the generated and real class conditional probability distributions of the i^{th} class, $\mathbf{h}$ and $G(\mathbf{z}|i)$, respectively represent the original and generated 3D-hyperspectral patches. Note that the aim of Generator (G) network is to fool the discriminator (O_{i6}). The purpose of the Discriminator (D) network is to classify the original patch as real (O_{i5}) and generated patch as fake (O_{i6}). The Classifier (M) network maximizes the class scores for i^{th} class w.r.t. the original patch, as well as the generated patch of i^{th} class (O_{i1} and O_{i3}) and at the same time minimizes the class scores w.r.t. the other classes (O_{i2} and O_{i4}).



Color	Land cover type	Samples
	Background	164624
	Asphalt	6631
	Meadows	18649
	Gravel	2099
	Trees	3064
	Painted metal sheets	1345
	Bare Soil	5029
	Bitumen	1330
	Self Blocking Bricks	3682
	Shadows	947
	Total samples	207400

Fig. 5. The ground truth, type associated with the land-cover classes, and the number of available samples in the University of Pavia (UP) dataset.



Color	Land cover type	Samples
	Background	309157
	Scrub	761
	Willow swamp	243
	CP hammock	256
	Slash pine	252
	Oak/Broadleaf	161
	Hardwood	229
	Swap	105
	Graminoid marsh	431
	Spartina marsh	520
	Cattail marsh	404
	Salt marsh	419
	Mud flats	503
	Water	927
	Total samples	314368

Fig. 6. The ground truth, type associated with the land-cover classes, and the number of available samples in the Kennedy Space Centre (KSC) dataset.

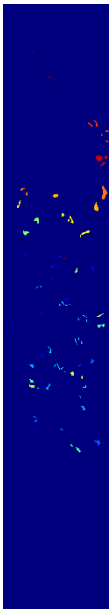
III. Experimental Results

The class imbalance is a long-standing problem in HSI classification, where classes do not have sufficiently equal numbers of representative samples to train the model. To show the effectiveness of the proposed method, we conduct classification experiments on four benchmark HSI datasets[1], including Indian Pines (IP), Kennedy Space Center (KSC), University of Pavia (UP), and Botswana (BW). The disjoint train-test samples are defined and given for IP, UP, and BW datasets, respectively. Figs. 4, 5, 6 and 7 show a detailed summary of these HSI datasets, including the ground truth, the type associated with the land-cover classes, and the number of available labeled samples.

A. Experimental Datasets

The Indian Pines (IP) dataset was gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [54] sensor over the Indian Pines test site in North-western Indiana. This

[1]http://dase.grss-ieee.org/



Color	Land cover type	Samples
	Background	374608
	Water	270
	Hippo grass	101
	Floodplain grasses1	251
	Floodplain grasses2	215
	Reeds1	269
	Riparian	269
	Firescar2	259
	island interior	203
	Accacia woodlands	314
	Accacia grasslands	248
	Short mopane	305
	Mixed mopane	181
	Exposed soils	268
	Total samples	377856

Fig. 7. The ground truth, type associated with the land-cover classes, and the number of available samples in the Botswana (BW) dataset.

dataset contains images of 224 spectral bands within a wavelength range of 400 to 2500 nm. The 24 null and corrupted bands have been removed. It consists of 16 mutually exclusive vegetation classes. The spatial dimension is 145×145 and some of its classes contain highly imbalanced samples as shown in Fig. 4.

The Kennedy Space Center (KSC) dataset was gathered in 1996 by AVIRIS [54] with wavelengths ranging from 400 to 2500 nm. The images have 512×614 pixels spatial dimension and 176 spectral bands after removal of some low signal-to-noise ratio (SNR) bands. The KSC dataset consists of in total 5202 samples of 13 upland and wetland classes.

The University of Pavia (UP) dataset was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [55] during a flight campaign over the university campus at Pavia, Northern Italy. The dataset contains images of 9 land-cover classes from urban areas and the spatial dimension is 610×340 pixels with 103 spectral bands in the wavelength range of 430 to 860 nm.

The Botswana (BW) dataset was collected by the UT center for space research during 2001 to 2004 and acquired by hyperion sensor on EO-1 over the Okavango, Delta, Botswana. The dataset contains images of 14 land-cover classes from seasonal swamps, occasional swamps and drier woodlands located in the distal portion of the Delta. The spatial dimension is 1496×256 pixels having 30 meters spatial resolution with 242 spectral bands in the wavelength range of 400 to 2500 nm. The uncalibrated and water corrupted 97 bands have been removed leading to remaining 145 spectral bands in the dataset.

B. Experimental Settings

In order to show and validate the effectiveness of the proposed `3D-HyperGAMO` network, employed to oversample minority classes while training for HSI classification, we

TABLE II
CLASSIFICATION RESULTS OF RF, MLR, SVM, GRU, LSTM, CNN1D, CNN2D, CNN3D AND 3D−HYPERGAMO ON DISJOINT TRAIN-TEST DATA FOR THE IP DATASET.

Class	RF	MLR	SVM	MLP	GRU	LSTM	CNN1D	CNN2D	CNN3D	Proposed
0	34.67	68.0	96.0	77.33	89.33	92.0	80.0	60.0	41.33	**97.33**
1	51.8	78.07	80.74	83.41	77.98	80.0	83.75	70.57	69.63	**89.83**
2	44.97	59.41	70.79	71.12	68.56	63.53	**73.76**	56.27	57.18	61.47
3	27.61	25.25	**51.52**	43.43	46.8	44.11	49.49	21.55	50.51	32.99
4	80.78	88.32	87.59	86.74	87.47	87.59	88.69	67.15	72.39	**88.93**
5	95.95	96.89	96.61	97.18	96.14	96.99	97.08	99.06	**99.34**	94.82
6	33.33	50.0	**100.0**	83.33	0.0	0.0	0.0	0.0	0.0	**100.0**
7	**99.87**	99.2	98.8	97.33	99.6	99.2	99.6	91.2	95.6	99.73
8	3.33	40.0	70.0	56.67	63.33	66.67	**83.33**	63.33	53.33	76.67
9	8.22	56.13	81.91	76.87	83.5	**85.88**	77.8	83.57	69.85	84.82
10	**89.77**	81.63	87.51	84.57	79.87	79.44	82.66	80.81	73.71	89.67
11	27.42	68.44	80.5	75.53	81.8	76.24	**83.92**	51.77	56.03	82.74
12	88.33	96.25	93.75	97.08	97.08	97.08	96.67	**100.0**	97.08	93.33
13	92.11	89.91	91.93	93.09	94.37	94.07	91.93	93.46	91.56	**97.13**
14	37.37	82.83	78.79	83.84	84.85	85.86	83.84	**90.24**	69.36	45.12
15	93.18	93.18	88.64	90.15	96.21	87.88	91.67	**98.48**	96.21	80.3
OA	65.9	78.15	85.08	83.91	83.36	82.96	84.62	77.65	75.09	**85.95**
AA	56.79	73.35	**84.69**	81.11	77.93	77.28	79.01	70.47	68.32	82.18
K(x100)	60.12	74.99	82.98	81.66	81.08	80.62	82.5	74.46	71.75	**83.99**

TABLE III
CLASSIFICATION RESULTS OF RF, MLR, SVM, GRU, LSTM, CNN1D, CNN2D, CNN3D AND 3D−HYPERGAMO ON DISJOINT TRAIN-TEST DATA FOR THE UP DATASET.

Class	RF	MLR	SVM	MLP	GRU	LSTM	CNN1D	CNN2D	CNN3D	Proposed
0	79.62	77.69	82.23	84.06	79.27	79.3	87.55	81.71	86.07	**90.09**
1	55.3	58.78	65.81	73.42	89.61	77.42	88.23	93.75	95.0	**95.01**
2	45.29	67.22	66.72	67.6	51.55	58.35	**75.81**	64.59	52.16	65.2
3	**98.68**	74.29	97.77	92.26	90.13	98.04	92.63	96.25	97.67	98.19
4	99.13	98.89	99.37	99.46	99.49	99.1	99.67	**99.79**	99.76	99.73
5	78.61	93.53	91.62	92.06	34.39	65.05	87.58	67.56	69.0	87.49
6	79.31	85.12	87.36	89.26	78.12	88.07	88.01	74.96	80.02	86.85
7	90.9	87.59	90.46	91.26	83.59	91.15	89.01	97.29	97.28	**97.61**
8	97.57	99.21	93.71	99.16	95.68	99.33	**99.33**	92.33	97.36	97.69
OA	70.14	72.23	77.8	81.46	79.59	79.39	88.41	87.69	88.88	**92.46**
AA	80.49	82.48	86.12	87.62	77.98	83.98	89.76	85.36	86.04	**90.87**
K(x100)	62.98	65.44	72.06	76.23	72.14	73.14	84.63	83.35	84.9	**89.87**

conduct a comparison that includes classical machine learning and representative deep learning methods available on [56] [2], such as random forest (RF) [57], multinomial logistic regression (MLR) [58], support vector machine (SVM) with radial basis function [59], gated recurrent unit (GRU) [60], long short term memory (LSTM) [61], CNN-1D [62], CNN-2D [56], and CNN-3D [23], [63] models. The performance of the proposed 3D−HyperGAMO is evaluated in terms of the widely used quantitative measurements, such as overall accuracy (OA), average accuracy (AA), and statistical kappa (κ) coefficient, respectively [64]. The ratio of correctly classified samples among the total test samples is determined as OA. The mean of class-wise accuracy is determined as AA. The κ represents a strong mutual agreement between the generated classification maps of one network model and the provided ground truth. The weights of the network 3D−HyperGAMO are randomly initialized and the parameters of the model are optimized through the Adam optimizer [65] using a learning rate of 0.0002 on Indian Pines, Kennedy Space Center, University of Pavia, and Botswana datasets, respectively. The spatial dimensions of noise vector is 100. To train the 3D−HyperGAMO model, each experiment was run for 8000 iterations till the generalization ability of the generated training samples in the minority classes is stabilized. In our experiments, the spatial window of size $11 \times 11 \times B$ are extracted from the pre-processed Indian Pines (IP), Kennedy Space Center (KSC), University of Pavia (UP), and Botswana (BW) datasets where B is set to 40. The computer equipment is composed by an Intel i9-9940X processor, 128GB of DDR4 RAM, NVidia Titan RTX with 24GB of DDR4 RAM. All the source code of the framework is implemented by using Keras library with TensorFlow as backend.

C. Classification Results over Hyperspectral Data Sets

The widely used disjoint train-test for IP data set provided by the IEEE GRSS DASE (http://dase.grss-ieee.org/) contains highly imbalanced training samples and is used to show the generalization ability of popular HSI classification methods. Fig. 4 shows 16 land-cover classes and the number of associated samples per class for the IP dataset. Table II tabulates the classification performance of all the compared methods which include overall accuracy (OAs), average accuracy (AAs), and Kappa coefficients, in addition to the class-specific accuracies for all the land cover classes of the IP dataset using several methods. The 16 rows in Table II represent the accuracy for each land-cover class, whereas the last three rows represent OA, AA, and Kappa values for all the compared methods. The best achieved classification results are represented in bold across all the compared methods. It can be observed that the proposed 3D−HyperGAMO model achieves best performance as compared to RF, MLR, SVM, GRU, LSTM, CNN1D, CNN2D, and CNN3D methods tabulated in Table II which shows the discriminative spectral and spatial feature learning capability of the 3-D convolutional layers of both the discriminator and the classifier. The proposed model achieves highest OA (85.95%) and Kappa coefficient (83.99%), while SVM shows the best AA (84.69%) performance for IP dataset. It is worth noting that the CNN1D and CNN2D both outperform the CNN3D when compared with CNN, whereas CNN1D achieves best OA (84.62%), AA (79.01%) and Kappa (82.50%) among these. To the best of the authors' knowledge, this is probably due to the presence of very similar textures over most of the spectral bands. Hence, due to the increased redundancy among the spectral bands, the CNN1D and CNN2D seem to outperform the CNN3D as observed in the experiments. Among the traditional classifiers RF, MLR, SVM, MLP, GRU, LSTM the best performance achieved is OA (85.08%), AA (84.69%) and Kappa (82.98%) for SVM. During the training phase, the newly generated samples can properly represent the characteristics of minority classes and provide helpful information to learn better the weights due to the combined efforts of the conditional feature mapping unit and the convex patch generator. The above observation may be one of the reasons behind the performance improvement using the proposed oversampling based HSI classification model. Additionally, the proposed 3D−HyperGAMO model shows a significant improvement of at least 0.87% in OA and 1.01% in Kappa values as compared to other methods listed in Table II.

To determine the generalization power of any well-known HSI classification method, the highly imbalanced UP disjoint train-test set is also adopted. Fig. 4 shows 9 land-cover classes and the number of samples belonging to each class for this dataset. The minimum number of samples are 947 present in the Shadows category while Meadows category contains the maximum number of samples (947) in the UP

[2]https://github.com/mhaut/hyperspectral_deeplearning_review

TABLE IV
CLASSIFICATION RESULTS OF RF, MLR, SVM, GRU, LSTM, CNN1D, CNN2D, CNN3D AND 3D-HYPERGAMO USING 5% RANDOMLY SELECTED TRAINING SAMPLES FROM THE KSC DATASET.

Class	RF	MLR	SVM	MLP	GRU	LSTM	CNN1D	CNN2D	CNN3D	Proposed
0	94.24	95.44	94.33	93.82	95.62	95.94	96.86	98.16	**98.8**	95.19
1	74.17	82.68	**92.21**	85.86	87.01	80.09	89.47	86.0	77.78	100.0
2	86.56	75.03	81.48	91.5	85.6	83.81	91.08	**96.02**	88.61	95.88
3	60.39	58.16	62.9	54.53	49.09	49.93	76.71	50.77	57.88	45.61
4	46.19	60.35	55.34	49.89	61.44	42.27	59.48	67.76	66.45	**92.16**
5	41.47	57.76	61.29	55.3	54.22	54.99	66.97	**94.47**	88.48	80.73
6	65.0	90.33	68.67	75.0	84.0	74.33	81.0	93.67	86.67	**71.00**
7	67.48	90.71	89.98	89.81	87.53	86.39	86.55	95.11	90.3	**98.04**
8	88.19	94.67	90.35	95.61	95.55	91.3	98.45	99.12	97.71	**99.80**
9	75.61	91.49	93.23	89.84	90.89	82.81	92.53	97.57	93.75	**100.0**
10	93.8	95.47	96.65	92.71	97.57	92.46	93.05	**99.58**	99.5	99.50
11	77.96	93.65	91.07	86.61	88.21	89.47	89.89	96.03	96.37	**100.0**
12	99.7	**100.0**	99.96	99.51	100.0	99.39	99.39	100.0	100.0	100.0
OA	82.01	89.07	88.94	87.78	88.55	85.93	90.92	94.04	92.48	**95.19**
AA	74.67	83.52	82.88	81.54	82.83	78.71	86.26	90.33	87.87	**90.94**
K(x100)	79.92	87.83	87.68	86.38	87.24	84.31	89.87	93.36	91.62	**94.64**

TABLE V
CLASSIFICATION RESULTS OF RF, MLR, SVM, GRU, LSTM, CNN1D, CNN2D, CNN3D AND 3D-HYPERGAMO USING 5% RANDOMLY SELECTED TRAINING SAMPLES FROM THE BOTSWANA DATASET.

Class	RF	MLR	SVM	MLP	GRU	LSTM	CNN1D	CNN2D	CNN3D	Proposed
0	99.61	**100.0**	99.87	100.0	100.0	99.48	100.0	98.96	99.35	99.22
1	86.81	96.18	96.18	95.83	89.93	82.99	80.56	95.83	98.26	99.31
2	92.58	98.18	95.24	96.78	97.34	97.06	**99.3**	98.88	97.9	97.48
3	86.11	92.49	91.99	91.67	93.95	95.26	98.2	99.18	91.67	100.0
4	75.88	82.9	87.08	80.94	78.65	77.29	84.6	82.16	82.37	**94.92**
5	48.29	69.36	66.49	77.71	75.69	71.84	74.55	**93.86**	88.01	93.62
6	91.73	95.8	96.48	95.8	95.39	95.66	97.43	99.32	99.73	99.86
7	83.94	98.1	91.02	97.93	98.62	94.13	99.65	97.41	96.72	**99.83**
8	77.85	83.67	80.65	86.13	85.68	79.31	88.93	98.1	92.51	**98.21**
9	75.56	87.29	85.73	82.06	82.63	83.62	90.4	**99.86**	94.77	99.44
10	88.28	92.53	90.34	93.33	91.26	92.53	89.89	96.44	98.74	**99.66**
11	87.98	92.25	92.25	93.8	95.93	93.8	94.77	99.81	97.87	**100.0**
12	77.12	92.55	89.93	85.62	89.8	83.53	93.33	96.86	**99.87**	99.87
13	96.67	96.3	97.04	84.07	98.15	95.19	87.41	81.48	95.19	**98.89**
OA	82.36	90.43	89.07	89.84	90.19	88.16	91.65	96.1	94.99	**98.45**
AA	83.46	91.25	90.02	90.12	90.93	88.69	91.36	95.58	95.21	**98.59**
K(x100)	80.89	89.63	88.16	89.0	89.37	87.17	90.95	95.77	94.57	**98.33**

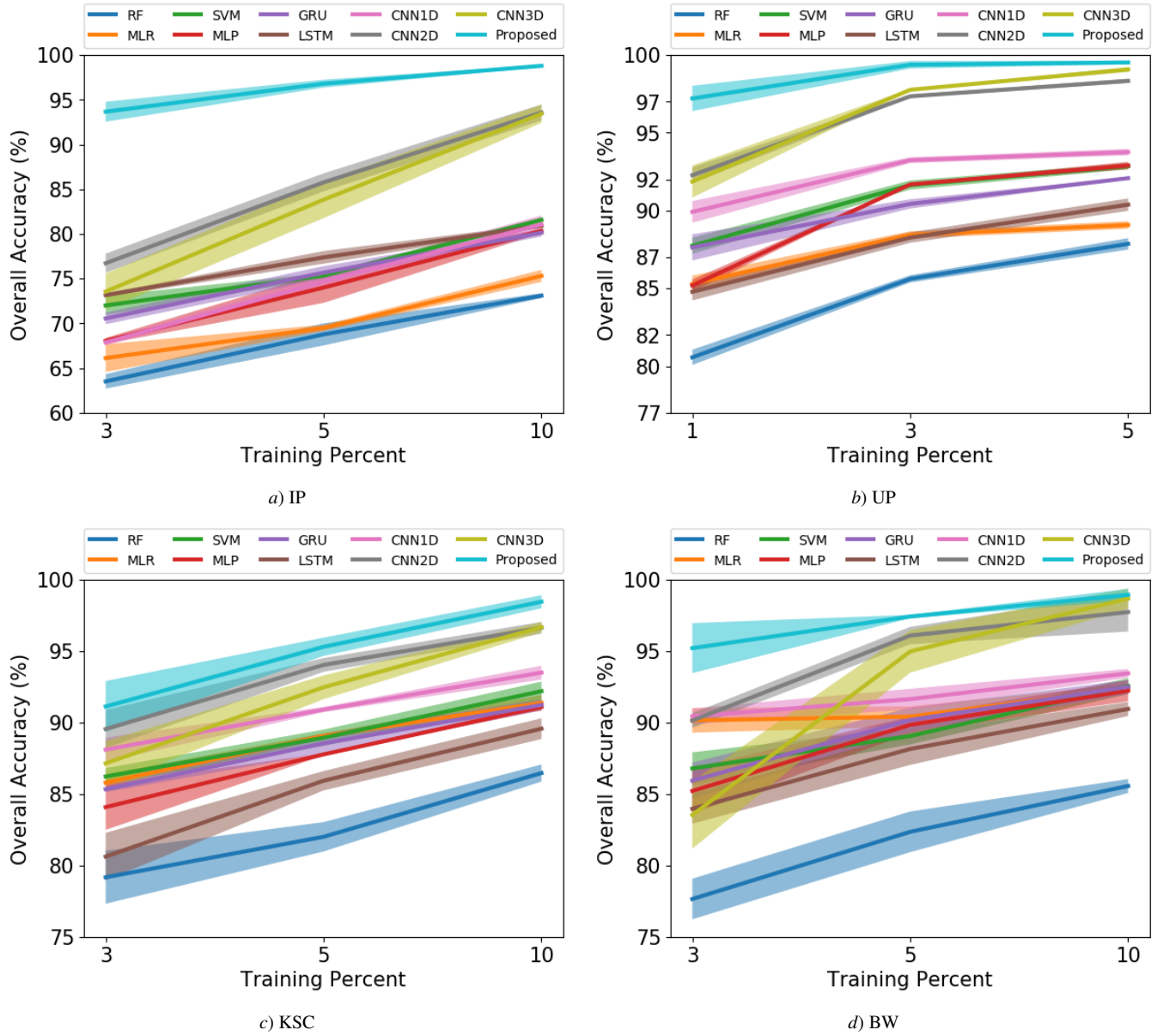
dataset. The statistical results of overall accuracy (OAs), average accuracy (AAs) and Kappa coefficients are shown in Table III including the class-specific accuracy for all the land cover classes of the UP dataset using state-of-the-art compared methods. Table III records the superior performance of the proposed 3D-HyperGAMO model in terms of the OA (92.46%), AA (90.87%) and Kappa (89.87) values among the compared methods. The best achieved class-specific classification accuracies are highlighted in bold. Over the UP dataset, the performance of the CNN3D is better and we observe that the second highest performance is obtained using CNN3D with OA (88.88%) and Kappa (84.90%) values among all the compared methods listed in Table III. The proposed 3D-HyperGAMO model achieves significant performance improvements of atleast 3.58%, 4.47% and 4.97% in terms of the OA, AA and Kappa when compared with the CNN3D reported in Table III. The SVM, a traditional classifier, achieves good performance in terms of OA (77.80%), AA (86.12%) and Kappa (72.06%) as compared to the RF (70.14%, 80.49% and 62.98%). It is also observed that the MLP performs quite well when compared with GRU and LSTM. Overall, the oversampling on minority classes facilitates the proposed model to learn more relevant features distributed across different categories.

The performance of the proposed 3D-HyperGAMO model is also evaluated using the KSC dataset also. The detailed summary of land-cover classes in KSC dataset is shown in Fig. 6. The number of class-specific samples and the difference between minimum and maximum class-wise samples show that the KSC dataset contains mild imbalances as compared to the IP and UP datasets. In order to show the effectiveness of the proposed 3D-HyperGAMO network, the labeled samples in the KSC dataset are randomly divided into 5% for training and 95% for testing, respectively. The statistical analysis in terms of the OA, AA and Kappa indexes of compared methods are reported in Table IV and the best achieved results are emphasized in bold. The 3D-HyperGAMO model obtains the highest classification accuracy in terms all three measures, i.e., the OA (95.19%), AA (90.94%) and Kappa (94.64%). Among the different CNN models, CNN2D shows the highest performance improvement of OA (1.56%), AA (2.46%) and Kappa (1.74%) over the CNN3D. The performance of SVM is always commendable when compared with classical machine learning based methods, i.e., RF, but its performance is a bit lower than that achieved by the MLR. Due to presence of less imbalance in the samples, MLP shows great improvement using OA, AA, and Kappa over GRU and LSTM. Moreover, the proposed network shows significant performance gain of at least 1.15% in OA, 0.61% in AA and 1.28% in Kappa values as compared to the methods listed in Table IV.

The performance of the proposed 3D-HyperGAMO network is also evaluated in the high resolution Botswana (BW) HSI dataset as well. Fig. 7 shows the description of the land-cover classes and the number of existing samples per class, which indicates imbalanceness for this dataset (as it was already the case with KSC). To test the performance of the proposed 3D-HyperGAMO network, the BW dataset is randomly divided into training and test sets, where 5% of the available samples are utilized for training and the remaining 95% are used for testing. Table V demonstrates the achieved class-wise accuracy and the statistical analysis of parameters OA, AA and Kappa for all the compared methods. The proposed network achieves the highest classification performance on the basis of the OA (98.45%), AA (98.56) and Kappa (98.33) as compared to the methods reported in Table V. The CNN2D performs well on BW dataset as compared to CNN1D and CNN3D, while achieves OA of 96.10%, AA of 95.58% and Kappa of 95.77% during the model evaluation. The performance achieved using classical classifiers show that the MLR performs quite well when compared with RF, SVM and MLP. It is worth noting that, among the recurrent classifiers, GRU exhibits better performance improvement as compared to LSTM. The proposed 3D-HyperGAMO model achieves a performance gain of at least 2.35% in OA, 3.01% in AA and 2.56% in Kappa values over BW dataset as compared to the methods listed in Table V.

D. Performance under Different Training Percentages

In order to prove the robustness and generalization ability of the proposed 3D-HyperGAMO network, it is important to justify the performance achieved on small and varying training sets. Our experiments are conducted over training sets with different samples. We select 3%, 5% and 10% randomly selected training sets from IP, KSC and BW datasets, whereas 1%, 3% and 5% randomly selected samples have been selected from the UP dataset for training. Fig. 8 illustrates the achieved overall accuracy (OA) against different training sizes using the



a) IP

b) UP

c) KSC

d) BW

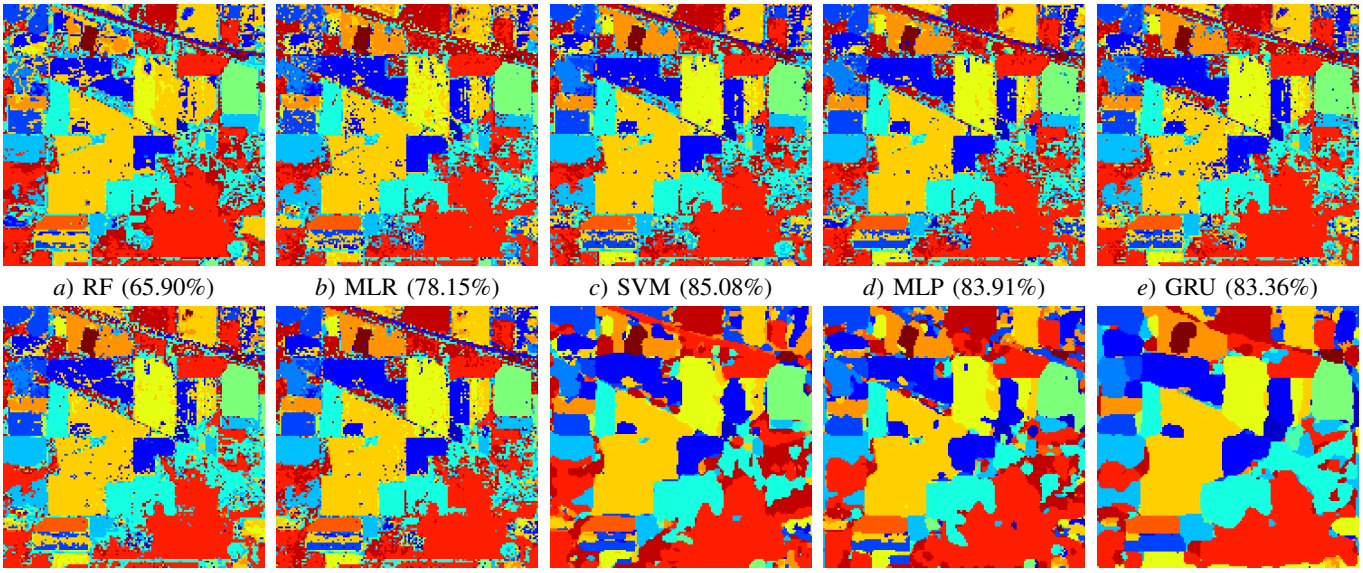
Fig. 8. Overall accuracy (OA) of different methods with different training data percentages (a) IP data set (b) UP data set (c) KSC data set (d) BW data set.

proposed and state-of-the-art compared methods over IP, UP, KSC and BW datasets. It can be easily seen that the gain in performance using the proposed method is more significant for the datasets with a higher degree of class imbalance.

Fig. 8(a) shows that the proposed `3D-HyperGAMO` network consistently outperforms the other compared methods using varying training samples, with significant differences for the IP dataset. The performance of CNN1D is not so good, but CNN2D achieves good performance as compared to CNN3D, obtaining similar performance for larger training sizes. Additionally, SVM, MLP, and LSTM exhibit similar performance when the training size increases, but the improvement in performance is not comparable for the RF. Fig. 9(a)-(h) shows the results of the classification methods: RF, MLR, SVM, MLP, GRU, LSTM, CNN1D, CNN2D, CNN3D and the proposed method for the IP data set. It can be observed that, even with very few samples, the proposed `3D-HyperGAMO` improves the obtained classification maps as compared to other methods.

Fig. 8(b) demonstrates the significant improvement in OA performance achieved by the proposed network as compared to the other tested methods when considering a small training set for the UP dataset. The CNN2D outperforms the CNN3D when the training size is very small, but once training size becomes more than 3% the CNN3D improves the performance. Among all the methods, RF exhibits the worst performance while the MLP and SVM achieve similar OA for larger training sizes. Similarly LSTM performs better than GRU.

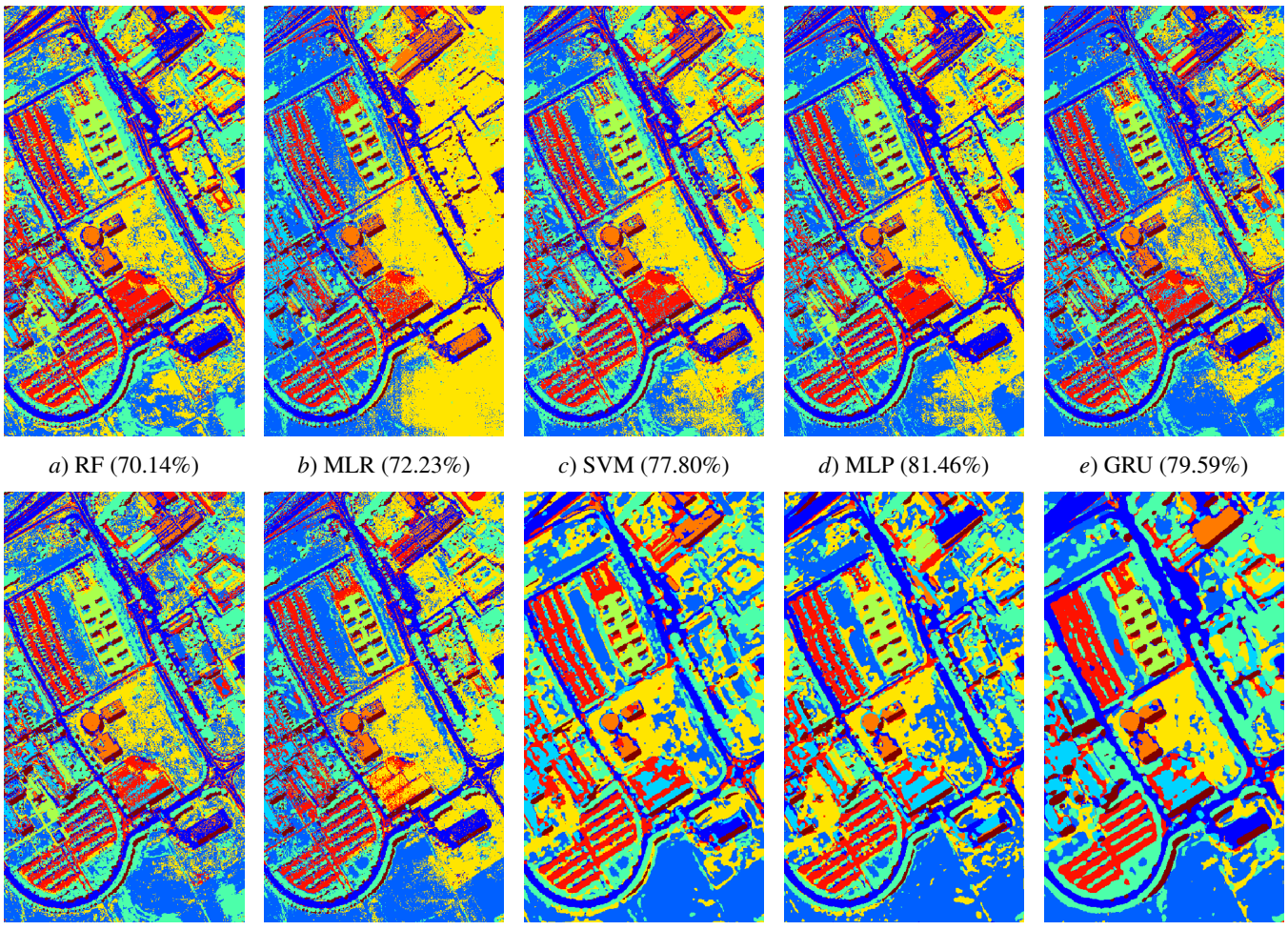
Fig. 8(c) illustrates the achieved performance in terms of OA (under varying training sizes) for the KSC dataset. It is evident that the proposed `3D-HyperGAMO` network smoothly outperforms the other obtained methods. In the beginning, the CNN2D performs well as compared to the CNN3D, but achieves similar OA when the size of the training set increases. The MLR and MLP both achieve very similar performance.



a) RF (65.90%) *b*) MLR (78.15%) *c*) SVM (85.08%) *d*) MLP (83.91%) *e*) GRU (83.36%)

f) LSTM (82.96%) *g*) CNN1D (84.62%) *h*) CNN2D (77.65%) *i*) CNN3D (75.09%) *j*) **Proposed (85.95%)**

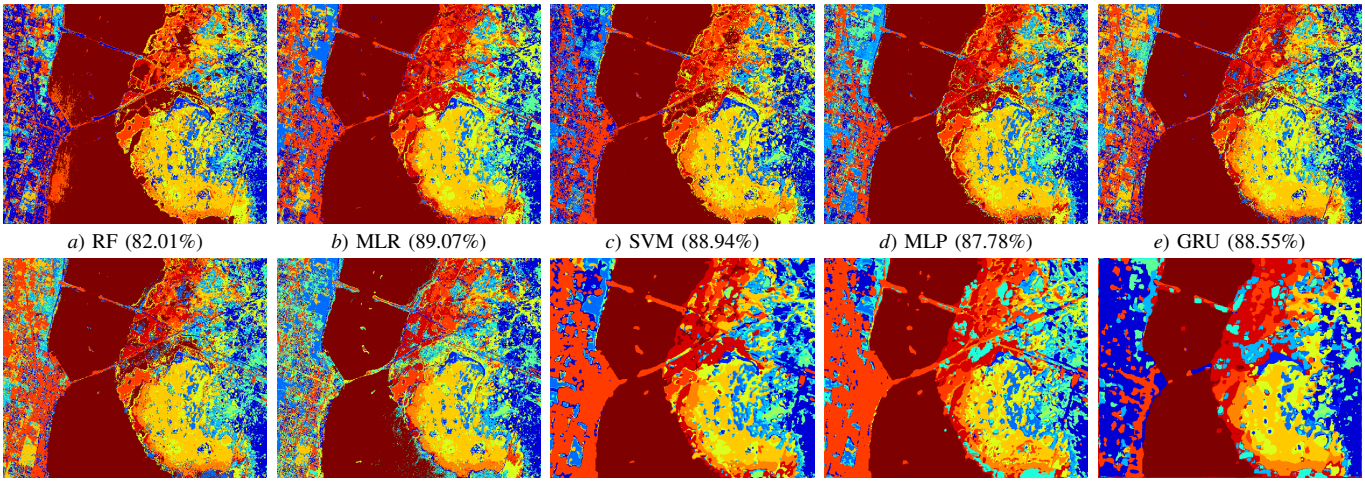
Fig. 9. Classification maps for the IP dataset obtained by (a) RF (b) MLR (c) SVM (d) MLR (e) GRU (f) LSTM (g) CNN1D (h) CNN2D (i) CNN3D and (j) 3D-HyperGAMO models.



a) RF (70.14%) *b*) MLR (72.23%) *c*) SVM (77.80%) *d*) MLP (81.46%) *e*) GRU (79.59%)

f) LSTM (79.39%) *g*) CNN1D (88.41%) *h*) CNN2D (87.69%) *i*) CNN3D (88.88%) *j*) **Proposed (92.46%)**

Fig. 10. Classification maps for the UP dataset by (a) RF (b) MLR (c) SVM (d) MLR (e) GRU (f) LSTM (g) CNN1D (h) CNN2D (i) CNN3D and (j) 3D-HyperGAMO.



a) RF (82.01%) *b*) MLR (89.07%) *c*) SVM (88.94%) *d*) MLP (87.78%) *e*) GRU (88.55%)

f) LSTM (85.93%) *g*) CNN1D (90.92%) *h*) CNN2D (94.04%) *i*) CNN3D (92.48%) *j*) **Proposed (95.19%)**

Fig. 11. Classification maps for the KSC data set by (a) RF (b) MLR (c) SVM (d) MLR (e) GRU (f) LSTM (g) CNN1D (h) CNN2D (i) CNN3D and (j) 3D-HyperGAMO models.

The LSTM achieves comparable performance in relation to the RF.

Fig. 8(d) illustrates the performance gained in terms of the OA under varying training sizes for BW datasets. The proposed `3D-HyperGAMO` network achieves outstanding performance when compared with other methods. Even though the CNN2D performs well for small training sizes, the best performance of the CNN3D can be clearly observed for a higher number of training examples. It is worth noting that the RF provides non-comparative OA results throughout the experiments. The MLR, MLP and SVM achieves similar performance of OA values with increasing training samples, but the worst OA is achieved by the LSTM.

E. Discussion of the Obtained Classification Maps

The robustness of any HSI classification method can also be verified through the quality of the generated classification maps. Figs. 9, 10, 11 and 12 show the obtained classification maps of the nine compared methods on the IP, UP, KSC, and BW data sets, respectively. As shown in Fig. 9(a)-(g) the maps for RF, MLR, SVM, MLP, GRU, LSTM, and CNN1D contain salt and pepper noise in the classification results due to the mis-classification of many points in the center of land-cover regions. This is clearly visible in the land-cover classes like Corn min, Corn, Hay windrowed and Building Grass Tree Drives, etc. in the generated map of the IP data set. As compared to the previous methods, the CNN2D and CNN3D obtain classification maps with higher quality, whereas the proposed `3D-HyperGAMO` model significantly boosts the performance through the generated classification maps. Moreover, it also helps to improve the uniformity of the land-cover regions shown in Fig. 9(h)-(j). The generated maps for UP dataset using RF, MLR, SVM, MLP, GRU, LSTM, CNN1D, CNN2D, CNN3D and `3D-HyperGAMO` methods are depicted in Fig. 10. The presence of similar spectral classes leads to increase the mis-classification probability. However, the proposed method successfully distinguishes the classes. Moreover `3D-HyperGAMO` achieves better delineation in almost all class boundaries as compared to other methods.

Fig. 11 gives a comparison of classification maps using various methods for KSC data set. Fig. 11(a)-(h) show that the obtained maps contain salt and pepper noise. As compared to CNN1D and CNN2D, CNN3D generates better classification maps. The proposed `3D-HyperGAMO` network obtains significant qualitative improvements of the obtained results when compared to other maps. A very similar trend is also observed in the BW dataset, as illustrated in Fig. 12. The maps generated using the proposed method are qualitatively better than those obtained by other methods.

IV. Conclusion

In this paper, we proposed a new `3D-HyperGAMO` framework that serves as an effective minority class over-sampling technique for handling the class imbalance in HSI data. It consists of a 3D-generator network which takes the 3D-HSI patches and a middle level representation of a noise vector produced by a conditional feature mapping unit to generate the new samples of that class. It also consists of a 3D-discriminator which distinguishes between the original and generated samples. In order to learn the class-specific information, a 3D-classification network is also used –which classifies the samples (original and generated) into the corresponding classes–. The number of samples to be generated is decided by the difference between the number of samples in the majority class and the corresponding minority class. The proposed method is tested on four benchmark HSI datasets. The classification results are compared with traditional and CNN-based methods. The proposed model outperforms existing models in most of the cases. Moreover, the performance gain due to the proposed model is more significant for small training sets. It can also be noticed that the performance of the proposed model is significantly improved if the degree of imbalance is higher. The classification maps generated using the proposed model are smoother and perceptually more appealing as compared to those obtained by the other tested models. Thus, from an

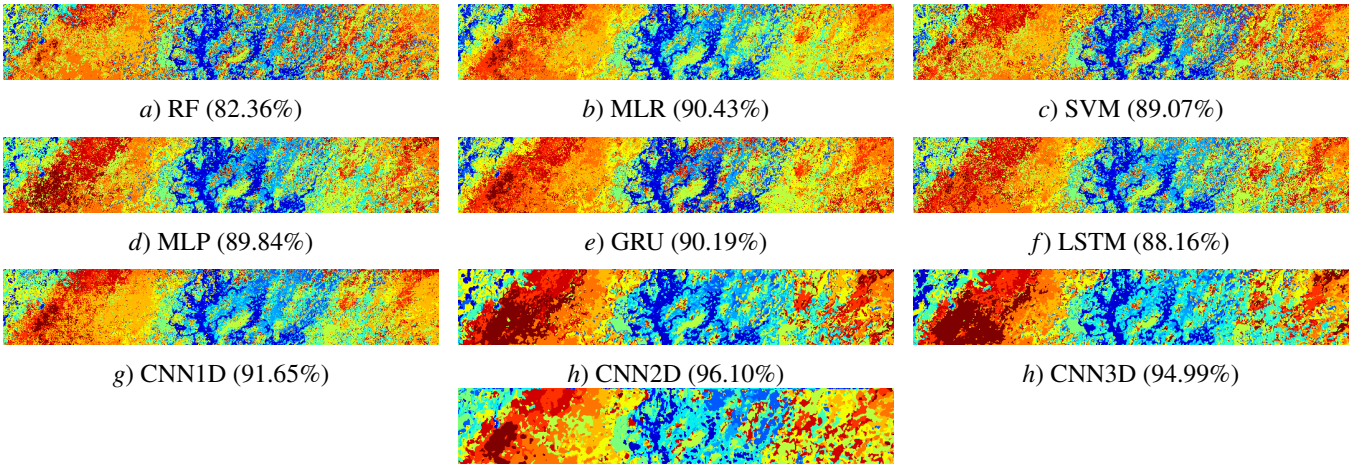


Fig. 12. Classification maps for the BOTSWANA dataset obtained by (a) RF (b) MLR (c) SVM (d) MLR (e) GRU (f) LSTM (g) CNN1D (h) CNN2D (i) CNN3D and (j) 3D-HyperGAMO models.

experimental viewpoint, we can conclude that the proposed `3D-HyperGAMO` is more effective with small training sets with high imbalance, which is the most common scenario in real remote sensing applications.

REFERENCES

[1] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple kernel learning for hyperspectral image classification: A review," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6547–6565, 2017.

[2] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[3] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE geoscience and remote sensing letters*, vol. 3, no. 1, pp. 93–97, 2006.

[4] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE transactions on geoscience and remote sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[5] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote sensing*, vol. 53, no. 3, pp. 1592–1606, 2014.

[6] B. Tu, X. Zhang, X. Kang, G. Zhang, J. Wang, and J. Wu, "Hyperspectral image classification via fusing correlation coefficient and joint sparse representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 340–344, 2018.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2018.

[10] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[11] N. He, M. E. Paoletti, J. M. Haut, L. Fang, S. Li, A. Plaza, and J. Plaza, "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 755–769, 2018.

[12] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.

[13] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 120–147, 2018.

[14] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[15] V. Rokhlin, A. Szlam, and M. Tygert, "A randomized algorithm for principal component analysis," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1100–1124, 2010.

[16] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844–853, 2016.

[17] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 11, pp. 6440–6461, 2018.

[18] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] X. Kang, B. Zhuo, and P. Duan, "Dual-path network-based hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, 2018.

[21] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3173–3184, 2018.

[22] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1–11, 2018.

[23] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Transactions on geoscience and remote sensing*, vol. 56, no. 8, pp. 4420–4434, 2018.

[24] M. He, B. Li, and H. Chen, "Multi-scale 3d deep convolutional neural network for hyperspectral image classification," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3904–3908.

[25] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[26] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 740–754, 2018.

[27] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep&dense convolutional neural network for hyperspectral image classification," *Remote Sensing*, vol. 10, no. 9, p. 1454, 2018.

[28] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 8065–8080, 2019.

[29] B. Fang, Y. Li, H. Zhang, and J. C.-W. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sensing*, vol. 11, no. 2, p. 159, 2019.

[30] Q. Xu, Y. Xiao, D. Wang, and B. Luo, "Csa-mso3dcnn: Multiscale octave 3d cnn with channel and spatial attention for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 1, p. 188, 2020.

[31] P. Wu, Z. Cui, Z. Gan, and F. Liu, "Residual group channel and space attention network for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 12, p. 2035, 2020.

[32] H. Gao, Y. Yang, D. Yao, and C. Li, "Hyperspectral image classification with pre-activation residual attention network," *IEEE Access*, vol. 7, pp. 176 587–176 599, 2019.

[33] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "Hybridsn: Exploring 3-d-2-d cnn feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, 2019.

[34] S. K. Roy, S. Chatterjee, S. Bhattacharyya, B. B. Chaudhuri, and J. Platoš, "Lightweight spectral–spatial squeeze-and- excitation residual bag-of-features learning for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5277–5290, 2020.

[35] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "Fpga: Fast patch-free global learning framework for fully end-to-end hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[36] Q. Liu, L. Xiao, J. Yang, and J. C.-W. Chan, "Content-guided convolutional neural network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[38] O. Tasar, S. Happy, Y. Tarabalka, and P. Alliez, "Colormapgan: Unsupervised domain adaptation for semantic segmentation using color mapping generative adversarial networks," *arXiv preprint arXiv:1907.12859*, 2019.

[39] J. Li, R. Cui, B. Li, R. Song, Y. Li, Y. Dai, and Q. Du, "Hyperspectral image super-resolution by band attention through adversarial learning," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[40] T. Jiang, Y. Li, W. Xie, and Q. Du, "Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[41] K. Jiang, W. Xie, Y. Li, J. Lei, G. He, and Q. Du, "Semisupervised spectral learning with generative adversarial network for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[42] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2017.

[43] Y. Zhan, K. Wu, W. Liu, J. Qin, Z. Yang, Y. Medjadba, G. Wang, and X. Yu, "Semi-supervised classification of hyperspectral data based on generative adversarial networks and neighborhood majority voting," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 5756–5759.

[44] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5329–5343, 2019.

[45] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-triplegan: Gan-assisted capsnet for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7232–7245, 2019.

[46] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[47] Z. Xue, "Semi-supervised convolutional generative adversarial network for hyperspectral image classification," *IET Image Processing*, vol. 14, no. 4, pp. 709–719, 2019.

[48] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE transactions on cybernetics*, 2019.

[49] J. Yin, W. Li, and B. Han, "Hyperspectral image classification based on generative adversarial network with dropblock," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 405–409.

[50] S. Pande, A. Banerjee, S. Kumar, B. Banerjee, and S. Chaudhuri, "An adversarial approach to discriminative modality distillation for remote sensing image classification," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[51] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Cva^{2}e: A conditional variational autoencoder with an adversarial training process for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[52] W.-Y. Wang, H.-C. Li, Y.-J. Deng, L.-Y. Shao, X.-Q. Lu, and Q. Du, "Generative adversarial capsule network with convlstm for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[53] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1695–1704.

[54] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 65, no. 3, pp. 227–248, 1998.

[55] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. Van der Piepen, and M. Schroder, "Rosis (reflective optics system imaging spectrometer)-a candidate instrument for polar platform missions," in *Optoelectronic technologies for remote sensing from space*, vol. 868. International Society for Optics and Photonics, 1988, pp. 134–141.

[56] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279 – 317, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271619302187

[57] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[58] J. Haut, M. Paoletti, A. Paz-Gallardo, J. Plaza, A. Plaza, and J. Vigo-Aguiar, "Cloud implementation of logistic regression for hyperspectral image classification," in *Proc. 17th Int. Conf. Comput. Math. Methods Sci. Eng.(CMMSE)*, vol. 3. Cádiz, Spain: Costa Ballena (Rota), 2017, pp. 1063–2321.

[59] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[60] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[61] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *The Journal of Supercomputing*, Feb 2020. [Online]. Available: https://doi.org/10.1007/s11227-020-03187-0

[62] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.

[63] 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 Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6808–6820, 2019.

[64] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote sensing of environment*, vol. 80, no. 1, pp. 185–201, 2002.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.