Hybrid 3D Convolution and 2D Depthwise Separable Convolution Neural Network for Hyperspectral Image Classification

Hüseyin Firat*, Mehmet Emin Asker and Davut Hanbay

Abstract— **Convolutional neural networks (CNNs) are one of deep learning methods that are often used to solve the problem of hyperspectral image classification (HSIC). CNN has a strong feature learning ability that can ensure more distinctive features for higher quality HSIC. The traditional CNN-based methods mainly use the 2D CNN for HSIC. However, with 2D CNN, only spatial features are extracted in HSI. Good feature maps cannot be extracted from spectral dimensions with the use of 2D CNN alone. By using 3D CNN, spatial-spectral features are extracted simultaneously. However, 3D CNN is computationally complex. In this study, a hybrid CNN method, which is a combination of 3D CNN and 2D CNN, is improved to solve the two problems described above. Using hybrid CNN decreases the complexity of the method compared to using only 3D CNN and can perform well against a limited number of training samples. On the other hand, in Hybrid CNN, depthwise separable convolution (DSC) is used, which decreases computational cost, prevents overfitting and enables more spatial feature extraction. By adding DSC to the developed hybrid CNN, a hybrid depthwise separable convolutional neural network is obtained. Extensive applications on frequently used HSI benchmark datasets show that the classification performance of the proposed network is better than compared methods.**

Index Terms—**3D Convolutional Neural Network, Depthwise Separable Convolution, Deep Learning, Hyperspectral Image Classificaiton**

I. Introduction

HYPERSPECTRAL IMAGES (HSIs) consist of tens or even hundreds of continuous narrow spectral bands with high spectral resolution, which can ensure abundant spatial-spectral feature information [1]. Since HSI has different spectral feature informations, it is commonly used in many fields such as agriculture, mining, astronomy, object tracking, military exploration, environmental monitoring, and vegetation [2-5]. In these applications, HSI is a major challenge for any classification method, as it includes multiple land cover classes resulting in high inter-class similarity and within-class variability. In recent years, many classification methods have been proposed to improve HSI classification (HSIC) performance. Since HSI contains rich spectral feature information, traditional classifiers such as support vector machines (SVM) [4], logistic regression [5], maximum likelihood [6], random forest [7] and k-nearest neighbors [5] have been proposed for HSIC. However, since these classification methods are based on spectral feature information, they cannot take full advantage of the important spatial feature information of HSIs. In this case, classification performance decreases. Nevertheless, traditional HSIC methods are based on handcrafted features with limited representation ability that do not fit well with the classification task [8].

In recent years, deep learning-based techniques have made great advances in many computer vision tasks, such as object detection [9], natural language processing [10], semantic segmentation [11-12], and image classification [13-14], because they can automatically extract robust and distinctive features from the original data in a hierarchical manner. Progress in deep learning techniques takes HSIC to a novel level. The training phase of deep learning techniques automatically extracts the features and uses these extracted features in the classification phase. Stacked autoencoder (SAE), a typical deep learning method, can extract spatial-spectral feature information. Later, it can combine this information for HSIC [15]. Chen et al. [16] applied the concept of deep learning to the field of HSIC at first. For spatial-spectral feature extraction and classification, they developed a method in which PCA, SAE and LR are used together. Tao et al. [17] proposed SAE based LR in which SAE and LR were used together to extract deep spectral features. Deep belief network (DBN) based classification is other deep learning methods used for HSI analysis. It has been proposed to combine the spatial feature information of HSI with spectral feature information [18]. Jia et al. [19] proposed a method in which PCA, hierarchical learning-based feature extraction and LR were used together for HSI analysis based on DBN. Li et al. [20] proposed a new HSIC method using optimal DBN and texture feature enhancement. While deep features can be extracted hierarchically in a layer-

HÜSEYİN FIRAT, is with Vocational School of Technical Sciences of Dicle University, Diyarbakır, Turkey,(e-mail: huseyin.firat@dicle.edu.tr).
https://orcid.org/0000-0002-1257-8518

MEHMET EMİN ASKER, is with Vocational School of Technical Sciences of Dicle University, Diyarbakır, Turkey, (e-mail: measker@dicle.edu.tr).
https://orcid.org/0000-0003-4585-4168

DAVUT HANBAY, is with Department of Computer Engineering of Inonu University, Malatya, Turkey, (e-mail: davut.hanbay@inonu.edu.tr).
https://orcid.org/0000-0003-2271-7865

Manuscript received December 21, 2021; accepted Jan 21, 2022.
DOI: 10.17694/bajece.1039029

based training process using SAE and DBN, training samples consisting of image patches to be input to these models need to be flattened to one dimension. In this case, spatial features in HSI cannot be fully used. Also, SAE and DBN are unsupervised learning algorithms and do not directly use informations of class while learning features [21]. Convolutional neural networks (CNN) are another deep learning method used for HSIC [22]. CNN has a strong feature learning ability that can ensure more distinctive features for higher quality HSIC [23]. The more distinctive the learned features are, the easier it will be to solve the classification problems [24]. This situation greatly improves the classification accuracy. Therefore, CNN-based methods used in HSIC are one of the most effective methods used to extract deeper spatial, spectral or spatial-spectral feature information. CNN-based methods are increasingly being applied as they effectively improve the HSIC performance. HSI is 3D data with two spatial dimensions (width, height) and spectral dimension (channel/depth). In CNN-based HSIC, mainly 2D CNN or 3D CNN is used for hierarchical feature extraction [25]. Chen et al. [26] proposed a 3D CNN method consisting of several convolution and pooling layers to extract spatial-spectral feature information. Also, dropout layer and L2 regularization were used to overcome the overfitting problem caused by the limited training samples. Roy et al. [27] proposed a hybrid spectral CNN (HybridSN) for HSIC. The HybridSN consists of using 3D CNN and 2D CNN together. 3D CNN extracts common spatial-spectral features from a spectral band stack. 2D CNN on top of 3D CNN extracts spatial features. Ahmad [28] proposed a fast 3D CNN method using spatial-spectral feature information to achieve better classification performances in HSIC. Ge et al. [29] proposed a deep network architecture based on multibranch feature fusion with 3D CNN and 2D CNN with different kernel size for HSIC. They also used Mish activation function instead of ReLU. He et al. [30] proposed a multiscale 3D deep CNN (M3D-DCNN) that can learn 2D multiscale spatial feature and 1D spectral feature jointly for HSIC. Mu et al. [31] proposed a multiscale and multilevel spatial spectral feature fusion network for HSIC. In the proposed method, From 3D to 2D alternative residual block combines spectral feature information extracted from 3D CNN with spatial feature information extracted from 2D CNN. Zhong et al. [32] proposed spectral-spatial residual network (SSRN) for HSIC. SSRN takes the 3D hyperspectral cube as input data without using any dimension reduction method. Residual blocks connect each 3D convolutional layer through identity mapping, which facilitates the back propagation of gradients. Mohan et al. [33] proposed a hybrid CNN model consisting of a multi-scale spatiospectral feature based 3D CNN and 2D CNN for HSIC. They used the hybrid dimension reduction method consisting of nonlinear kernel principal component analysis (KPCA) and linear gaussian random projection (GRP) to reduce spectral band redundancy and provide optimum band extraction. Mohan et al. [34] proposed a hybrid CNN architecture consisting of 3D CNN and 2D CNN for HSIC. The proposed method uses PCA as the preprocessing step. 3D CNN is used to extract spatial-spectral features simultaneously. Then 2D CNN is used to extract more spatial features. However, the redundancy in its spatial features is decreased by octave convolution instead of classic convolution. Cao et al. [35] proposed a deep 3D-2D-SSHDR method for HSIC that continuously extracts spectral and spatial feature information by combining spectral residual blocks, spatial hybrid dilated convolutions residual blocks, and outermost skip connection. Hamida et al. [36] and Li et al. [21] designed a 3D CNN to simultaneously extract spatial-spectral feature information for HSI classification.

In recent years, there has been an intense development of CNN-based methods for HSIC in which spatial features are adapted by a 2D CNN method. Nevertheless, since only the spatial information in the HSI will be preserved with 2D CNN, good distinguishing feature maps cannot be extracted from spectral dimensions. That is, if 2D convolution processes are implemented to HSIs, important spectral information is lost. By using 3D CNN, this problem is overcome. Since spectral and spatial features are extracted simultaneously with 3D CNN, the structural features of 3D HSIs are fully utilized in this study. However, the use of 3D CNN alone is computationally complex. Therefore, in the proposed method, 3D CNN and 2D CNN are joined to eliminate the shortcomings of 3D CNN and 2D CNN methods. In this way, maximum classification accuracy is achieved by fully utilizing both spatial and spectral feature maps. Also, some of the traditional 2D convolution layers have been replaced by depthwise separable convolution (DSC) layers. The purpose here is to solve the parameter and operating costs problem and prevent overfitting. In this study, a hybrid method consisting of 3D CNN, 2D CNN and 2D DSC layers is proposed. First of all, PCA is applied to the 3D HSI as a preprocessing step. Then, 3D hypercube datas are obtained to be given to the input of the network by neighborhood extraction. Firstly, 3D CNN is applied to extract spatial-spectral features simultaneously. Then feature information is converted from 3D data to 2D feature map and 2D CNN is applied to extract spatial feature. Finally, more spatial features are extracted using 2D DSC layers. DSC can improve HSI's feature learning capability and decrease computational complexity.

The rest of this paper is organized as follows: The datasets used in the study with PCA, 3D/2D Convolutional neural network, Depthwise separable convolution which constitute the theoretical background of the proposed method in this study, are explained in Section 2. In addition, the proposed method is detailed in Section 2. Application results and discussions about results are given in Section 3. This study is summarized in Section 4.

II. MATERIALS AND METHODS

A. *Principal Component Analysis (PCA)*

PCA is the most commonly used method in dimension reduction. With PCA, a significant reduction in the number of spectral features is possible while preserving most of the spatial features included in the HSI dataset. Significant correlation between spectral bands in HSI forms the principal of PCA. Analysis with PCA defines the optimal linear combination of the original spectral bands that takes into account the change of pixel values and also attempts to remove the correlation between spectral bands in HSI. The mathematical basic of PCA is based on the eigenvalue decomposition of the covariance matrix of the HSI spectral bands. Pixels of HSI are adjusted as a vector whose dimension is the same as the number of spectral bands. $X_i = [x_1, x_2, x_3, \dots \dots, x_N]^T$, N is the number of HSI spectral bands. The mean of all pixel vectors is calculated as in Equation (1).

$$m = \frac{1}{M}\sum_{i=1}^{M}[x_1, x_2, x_3, \dots \dots, x_N]_i^T \qquad (1)$$

$M = p * q$ represents the number of pixel vectors for an HSI consisting of "q" columns and "p" rows. The covariance matrix is defined as in Equation (2).

$$C = \frac{1}{M}\sum_{i=1}^{M}(X_i - m)(X_i - m)^T \qquad (2)$$

The eigen decomposition of the covariance matrix, that is, the eigen values together with their corresponding eigen vectors, is expressed as in Equation (3).

$$C = ADA^T \qquad (3)$$

D is the diagonal matrix consisting of the eigenvalues $(\lambda_1, \lambda_2, \dots \dots, \lambda_N)$ of C, and $A = a_1, a_2, a_3, \dots \dots, a_N$, is the orthogonal matrix with corresponding eigenvectors (each of dimension *N*) as columns [37]. The linear transformation $y_i = A^T X_i, i = 1,2, \dots \dots, M$ is adapted to obtain changed pixel vectors, which are PCA-converted spectral bands of the original HSI. The first row K of the A^T matrix is chosen so that the rows are eigenvectors suitable to their eigenvalues arranged in a decreasing order. The selected K rows are multiplied by the pixel vector X_i to give PCA spectral bands consisting of most of the information included in the HSI spectral bands [38].

B. 3D/2D Convolutional Neural Network

Convolutional neural network (CNN), which is one of the deep learning-based methods in HSIC, has been commonly used recently. CNN is a deep artificial neural network consisting of convolution, pooling, flatten, fully connected and softmax layers. The convolution layer is the first layer of CNN used to extract features by applying convolution operation to the input data. The convolution operation is performed by the inner product of the input image matrix and the kernel or a filter matrix. In this way, the output (feature map) of the input image is formed by shifting the kernel matrix over the input image matrix. The pooling layer reduces the width and height of the output (feature map) obtained as a result of the convolution operation without losing its features. In this way, both the required processing power is reduced and the unnecessary features that are caught are ignored and more important features are focused on. There are two different pooling methods generally used in CNN. Maximum and average pooling. In the pooling layer, which has a kernel (filter) as in the convolution layer, the kernel moves over the input image matrix. But instead of the convolutional operation, it applies the determined pooling method. In other words, if you are applying max pooling, it takes the largest value in the area covered by the filter, and if you are applying average pooling, it takes the average of the values in the filter. This reduces the spatial dimension and retains important features. The flatten layer prepares the data at the input of the last and most important layer, the fully connected layer. Generally, neural networks take input data from a one-dimensional array. The data in this neural network is the one-dimensional array of matrices from the convolutional and pooling layers. The fully connected layer receives the data from the flatten layer and performs the learning process through the neural network. Activation function used as the last layer for classification in deep learning methods is softmax function. As a result of the softmax function, the probabilistic distributions of the classes to be used in the classification process are generated. The probabilistic value generated for each class is in the range of 0 to 1. The total probability value of all classes is equal to 1 [33]. For x inputs, the output of a single neuron is calculated as in Equation (4).

$$t = f(w * x + bias) \qquad (4)$$

In Equation (4), w represents the filter weight. $f\,(.)$ denotes nonlinear activation function implemented to a weighted input sum. In the 2D CNN method, convolution is performed using the 2D kernel before passing the input HSI data through the activation function. The convolution performs by calculating the sum of the inner product between kernel and the input HSI. The kernel moves on on the HSI to cover the entire spatial dimension. This convolution operation provides to extract spatial features from the HSI. The 2D convolution output of each neuron is formulated as in Equation (5).

$$t_{mn} = f\left(\sum_{l}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1} k_{ij}x_{(i+m)(j+n)} + bias_{mn}\right) \qquad (5)$$

In Equation (5), t_{mn} is the feature extracted at *(m,n)* position. k is the 2D convolution kernel at dimension $h \times w$. In the case of a 2D image, this convolution operation is applied on all feature maps *(l)* in the receiver area and sums all values for non-linear activation. This process is repeated for all layers in the case of multi-dimensional data. When data are 3D they have spatial-spectral dimensions. 2D convolution fails for this input data. Because, in traditional 2D CNN, convolution processes are applied only to 2D feature maps that capture features in spatial dimension. 3D CNN is a changed version of 2D CNN methods that apply 3D convolution instead of 2D convolution in 2D CNN. When convolution processes are applied to 3D data, it is desirable to capture features from both spatial and spectral dimensions. For this purpose, 3D CNN is used, in which 3D convolution processes are implemented to 3D hypercubes to calculate co-spatial spectral features from 3D input data. 3D convolution enables the extraction of spatial-spectral features from 3D images. The feature extracted from the 3D CNN method is formulated as in Equation (6).

$$t_{mnd} = f\left(\sum_{l}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}\sum_{r=0}^{b-1} k_{ijr} x_{(i+m)(j+n)(r+d)} + bias_{mnd}\right) \tag{6}$$

In Equation (6), t_{mnd} is the feature extracted at *(m,n,d)* position. b is the dimension of the 3D kernel along the spectral dimension. The kernel (k) is 3D and the features are calculated by applying 3D convolution on the 3D input data. In traditional 2D CNNs, the convolution process to obtain 2D feature maps is applied only on spatial dimensions and covers all feature maps of the previous layer. However, for HSIC, it is desirable to obtain spectral features information as well as spatial features. 2D CNNs cannot extract spectral features. 3D CNN kernel can simultaneously extract spatial-spectral feature information from HSI despite increasing computational complexity. To benefit advantage of the automatic feature learning of both 3D CNN and 2D CNN, a hybrid CNN method consisting of 3D and 2D CNN is used under the HSIC.

C. Depthwise Separable Convolution

HSIs are different from 2D images and include a lot of information in the spatial dimension. Spatial features are determined by the excess information contained in the spatial dimension of HSIs. These spatial features can compensate for the shortcomings of spectral features to improve the model's ability to capture features. Spatial features of HSIs can be extracted with 2D CNN. However, since HSIs are 3D, they are not suitable for simultaneous extraction of spatial-spectral features with 2D CNN. Considering of the inadequate use of hyperspectral data information by 2D convolution, a DSC layer is added after the 2D convolution layer, which can increase spatial features and decrease parameters. With the addition of the DSC layer, a lot of spatial-spectral features are extracted to enable the method to losslessly distinguish spatial information of different spectral bands [39].

DSC consists of two separate operations, namely, depthwise convolution (DC) and pointwise convolution (PC) (1x1 convolution or also known as point-by-point convolution). As shown in Fig. 1, a separate convolution operation is performed on each channel of the input image in DC. With DC convolution operation, spatial features are extracted on each dimensions. In PC convolution operation, 1x1 standard convolution is applied to the depthwise feature map obtained as a result of DC convolution. PC convolution is used to assemble the feature map across channels [40].

Fig. 1a shows the standard convolution operation and Fig. 1b shows the DSC. In Figure 1, the size of the 3D input image is WxHxC. W is the width of the input image, H is the height of the input image, and C is the number of depth/channel. Suppose the size of the convolution kernel is KxKxC and the number of convolution kernels is N. The input image is convoluted with each convolution kernel and a feature map (output) of size WxHxN is obtained. The computational cost of the standard convolution operation is as in Equation (7).

$$cost_{standart\_conv} = KxKxCxNxWxH \tag{7}$$

The DSC implements a single channel filtering operation for each input channel, added after the DC filtering, and then performs PC. In DCS, in order to apply DC to the input image of size WxHxC first, it is divided into C groups with convolution kernel size KxKx1. Each group then performs standart convolution, which is equivalent to extracting the spatial features of each input channel C, ie Depthwise feature. The output size obtained after this operation is WxHxC. In PC, on the other hand, 1x1xC convolution kernels are used. In other words, 1x1xC convolution operation with N number of convolution kernels is applied to the output feature map obtained at the end of DC. The size of the output feature map after DC and PC is WxHxN. Equations (8) and (9) give the computational costs of DC and PC, respectively.

$$cost_{DCconv} = KxKxWxHxC \tag{8}$$

$$cost_{PCconv} = CxWxHxN \tag{9}$$

The calculation required to compare the DSC with the standard convolution is given in Equation (10).

$$\frac{cost_{DCconv} + cost_{PCconv}}{cost_{standart\_conv}} = \frac{1}{K^2} + \frac{1}{N} \tag{10}$$

Considering Equation (10), it can be seen that the trainable parameters and calculations of the DSC are $\frac{1}{K^2} + \frac{1}{N}$ times of the standard convolution. With the use of DSC, the number of trainable parameters and the computational cost are significantly reduced.

D. Information of HSI Datasets

Three HSI datasets including Indian Pines, Pavia of University and Salinas datasets were used to evaluate the performance of the proposed method and state-of-the-art techniques. Indian Pines (IP), Pavia of University (PU) and Salinas (SA) datasets are taken from the website http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

The IP is data obtained by the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) sensor at the Indian Pines test site in North-western Indiana. Each spectral image in this dataset is 145 × 145 spatial dimension. The sensor achieved a total of 224 spectral bands in the 0.4-2.5 micrometers wavelength range. From these 224 bands, 24 bands that are in the water absorption region and not useful were removed. A total of 200 spectral bands are used in applications. The IP includes 16 classes and 10,249 samples. The PU is data obtained by Reflective Optics System Imaging Spectrometer (ROSIS) optical sensors over the Pavia University in northern Italy. PU is 340 pixels width and 610 pixels height, with a spatial resolution of 1.3 meters per pixel. The dataset has a 115 wavelength ranging from 0.43-0.86 micrometer. After removing 12 noise bands in applications, a total of 103 spectral bands are used. The PU includes 9 classes and 42,776 samples. The SA is data collected by the AVIRIS sensor over Salinas Valley region of California. SA has a spatial resolution of 3.7 meters per pixel. The dataset is a remote sensing image 217 pixels width and 512 pixels height and contains 224 spectral bands. From these 224 bands, 20 bands that are in the water absorption region and not useful were removed. A total of 204 spectral bands are used in applications. The SA includes a total of 16 classes and 54,129 samples. The class and sample number information of the datasets are given in Table I, while the false-color images are shown in Fig. 2.

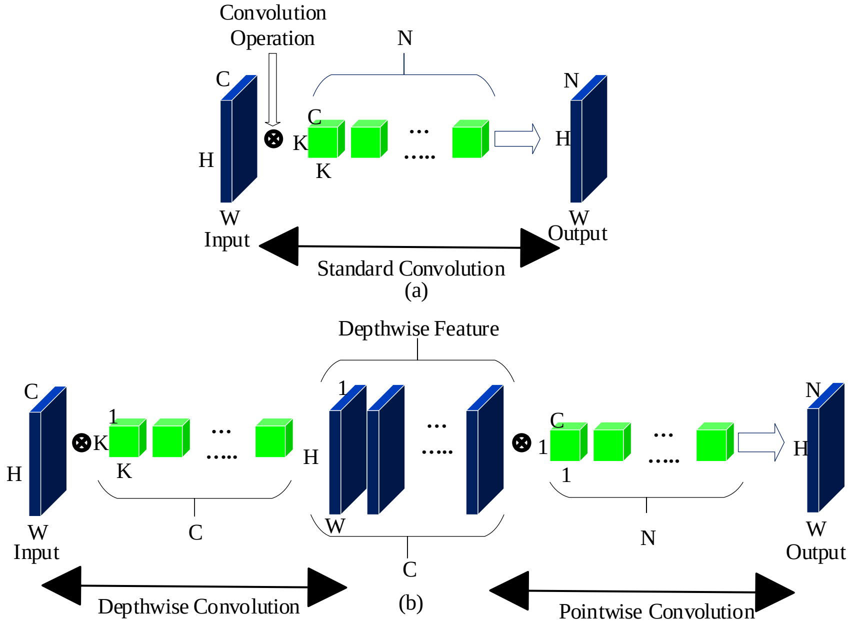

Fig. 1. (a) Standart convolution operation. (b) Depthwise separable convolution operation

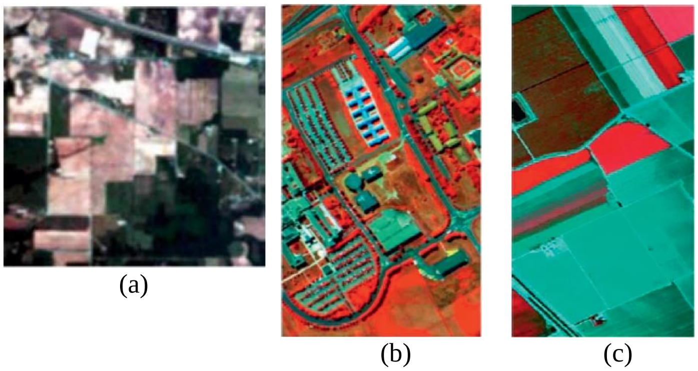

Fig. 2. False-color images of (a) Indian Pines, (b) University of Pavia, (c) Salinas datasets

TABLE I
INFORMATION REGARDING SAMPLES OF EACH CLASS IN IP, PU, AND SA DATASETS

No	Indian Pines (IP)		Salinas (SA)		Pavia University (PU)	
	Classes	Samples	Classes	Samples	Classes	Samples
1	Alfalfa	46	Brocoli_green_weeds_1	2009	Asphalt	6631
2	Corn-notill	1428	Brocoli_green_weeds_2	3726	Meadows	18,649
3	Corn-mintill	830	Fallow	1976	Gravel	2099
4	Corn	237	Fallow_rough_plow	1394	Trees	3064
5	Grass-pasture	483	Fallow_smooth	2678	Painted metal sheets	1345
6	Grass-trees	730	Stubble	3959	Bare soil	5029
7	Grass-pasture-mowed	28	Celery	3579	Bitumen	1330
8	Hay-windrowed	478	Grapes_untrained	11,271	Self-blocking Bricks	3682
9	Oats	20	Soil_vinyard_develop	6203	Shadows	947
10	Soybean-notill	972	Corn_senesced_green_weeds	3278		
11	Soybean-mintill	2455	Lettuce_romaine_4wk	1068		
12	Soybean-clean	593	Lettuce_romaine_5wk	1927		
13	Wheat	205	Lettuce_romaine_6wk	916		
14	Woods	1265	Lettuce_romaine_7wk	1070		
15	Buildings-grass-trees-drives	386	Vinyard_untrained	7268		
16	Stone-steel-towers	93	Vinyard_vertical_trellis	1807		
Total Number		10,249		54,129		42,776

E. Our Proposed Method

HSI X is specified as a 3D data cube with two spatial and one spectral dimensions of size $MxNxD$. X is the original input. While D indicates the number of spectral bands, M and N are the spatial width and height of the HSI. The ground truth of the input image Y is converted using one-hot encoding and expressed as $Y = (y_1, y_2, \dots \dots, y_C)$. C indicates the number of classes in the HSI. HSI pixels exhibit high inter-class similarity, high intra-class variability, overlapping and nested regions, which require extensive effort for any classification method. To overcome these problems, it is necessary to remove the spectral band redundancy. As a preprocessing step to remove spectral band redundancy, conventional principal component analysis (PCA) is first applied to the original HSI (X) data along the spectral bands. With PCA, the number of spectral bands is reduced from D to B while maintaining the same spatial dimensions (i.e., width M and height N). That is, using PCA, the spatial feature information, which is very important for recognizing any class, is preserved, while only the number of spectral bands is reduced. Thus, it is reduced to the desired number of bands. After the PCA, the modified input image X is represent as $MxNxB$. M is the width, N is the height, and B is the number of newly obtained spectral bands. In the second stage, to use HSIC techniques, the HSI cube is split into small overlapping 3D spatial patches from which real labels based on the central pixel are generated. 3D neighboring patches $SxSxB$ is created from the HSI cube (X) centered in the spatial position (a, b) and covering the SxS window size or spatial dimension and all spectral bands B. The total number of 3D patches (n) produced from the HSI cube (X) is found by $(M - S + 1)\, x\, (N - S + 1)$. Thus, these patches at location (a, b), covers the width from $a - (S - 1)/2$ to $a + (S - 1)/2$, height from $b - (S - 1)/2$ to $b + (S - 1)/2$, and all spectral bands (B) of HSI cube (X) [27][28]. The generated 3D HSI data cubes is given to the input of the Hybrid 3D/2D CNN architecture for feature extraction and classification.

The proposed Hybrid 3D/2D CNN consists of 3 3D CNNs, 2 2D CNNs, 1 Separable Conv layers (DC + PC layers), flattening, 2 Fully connected (FC), dropout and softmax layers. To preserve the maximum spatial-spectral information, the pooling layer in CNN is not to be used. The 3D convolution process is applied three times to simultaneously increase the number of spatial-spectral feature maps and can protect the spectral information of the input HSI data in the output volume. Dimensions of the applied 3D convolution kernels and the filters used are as follows. In the first convolutional layer, 32 filters with the dimension of 3x3x7 (ie, two spatial and one spectral dimensions), 64 filters with the dimension of 3x3x3 in the 2nd convolution layer, 64 filters with the dimension of 1x1x1 in the 3rd convolution layer are used. 2D convolution is applied twice before the flattened layer, bearing in mind that it strongly discriminates spatial information in different spectral bands without significant loss of spectral information, which is crucial for HSI data. The dimensions of the 2D convolution kernels and the filters used are as follows. 128 filters with the dimension of 3x3 in the 1st convolution layer and 128 filters with the dimension of 1x1 in the 2nd convolutional layer are used. Similarly, 2D DSC layers containing depthwise and pointwise convolution layers before the flattening layer are applied. The dimensions of the 2D DSC layer kernels and the filters used are as follows. 128 filters with the dimension of 3x3 in the DSC layer are used. Since $f(.)$ ReLU is the most frequently used activation function in CNN, it is used in all convolution layers. ReLU is faster than other activation functions when using gradient descent techniques for training. ReLU is formulated as in Equation (11).

$$ReLU(t) = max(0, t) \tag{11}$$

The features extracted after the 3D and 2D convolution layers are flattened and given as input to the FC layers for classification. In proposed method, two FC layers with 256 and 128 neurons are used. To prevent overfitting, a dropout layer with a dropout rate of 0,4% is implemented after each FC layer. The output of the FC layer is given to the simple softmax classifier to produce the classification result. Softmax is another activation function often used in the last layer of a deep learning method for classification. This activation function produces the probability distribution of all samples and their sum equals one. More details on the proposed method are shown in Table II. The total number of trainable weight parameters in the proposed method for PU dataset is 1,465,481.

TABLE II
SUMMARY OF THE PROPOSED METHOD FOR PU

Layer(Type)	Output Shape	Parameters
Input (InputLayer)	(11, 11, 15, 1)	0
Conv3d	(9, 9, 9, 32)	2048
Conv3d_1	(7, 7, 7, 64)	55360
Conv3d_2	(7, 7, 7, 64)	4160
Reshape	(7, 7, 448)	0
Conv2d	(5, 5, 128)	516224
Separable_conv2d	(5, 5, 128)	17664
Conv2d_1	(5, 5, 128)	16512
Flatten	3200	0
Dense	256	819456
Dropout	256	0
Dense_1	128	32896
Dropout_1	128	0
Dense_2	9	1161
Total Trainable Parameters		1,465,481

III. APPLICATION RESULTS AND DISCUSSIONS

All applications are performed using Colaboratory (Colab) Notebook. Colab is an online platform offered by Google that provides free access to the Graphical Processing Unit (GPU) and Tensor Processing Units (TPU) as hardware accelerators. Google Colab needs a good internet speed while running on any hardware accelerator. It is preferred to use the TPU as a hardware accelerator while running our applications. The TPU offers the ability to run code on a Python 3 desktop computer with 35 GB of RAM and 107.77 GB of storage for data computing. In all applications, categorical crossentropy with 0.001 learning rate and Adam optimizer with 1e-06 decay function used for optimization is used. The training process repeats for 256 batch sizes and 100 epochs. ReLU is used as activation function in all layers except the Softmax layer. The proportion of training samples and the size of the input are factors that affect the accuracies of the HSIC. All three datasets use 11 × 11 neighborhoods (window size) for convolution. Training sample rates of IP, PU and SA datasets are taken as

20%, 10% and 10%, respectively. On the other hand, in our applications, 30, 15 and 15 principal components are used for IP, PU and SA datasets, respectively. In order to make a better comparison with other deep learning-based methods, the PC values were chosen in this way. That is, dimension of the 3D patches obtained as a result of neighborhood extraction was set to 11 × 11 × 30 for IP, 11 × 11 × 15 for PU and SA datasets, respectively. Overall accuracy (OA), Average accuracy (AA) and Kappa statistics (K) have used to evaluate the classification results of each method. OA is computed by the ratio between correctly classified samples in the test samples and the total number of test samples (Eq. (12)). AA is the average value of the accuracy of each class (Eq. (13)). K is a statistical measurement metric that provides mutual information regarding a strong agreement between the basic truth map and the classification map (Eq. (14)). The advantage of using the Kappa value is to consider the effect of uncertainties on classification accuracy when predicting classification accuracy.

$$OA = \sum_{i=1}^{K} M_{ii} \,/ N \tag{12}$$

$$AA = \sum_{i=1}^{K}\left(M_{ii} \middle/ \sum_{j=1}^{K} M_{ij} \right) \middle/ K \tag{13}$$

$$Kappa = \left(N\sum_{i=1}^{K} M_{ii} - \sum_{i=1}^{K}(M_{i+}\ x\ M_{+i}) \right) \middle/ \left(N^2 - \sum_{i=1}^{K}(M_{i+}\ x\ M_{+i}) \right) \tag{14}$$

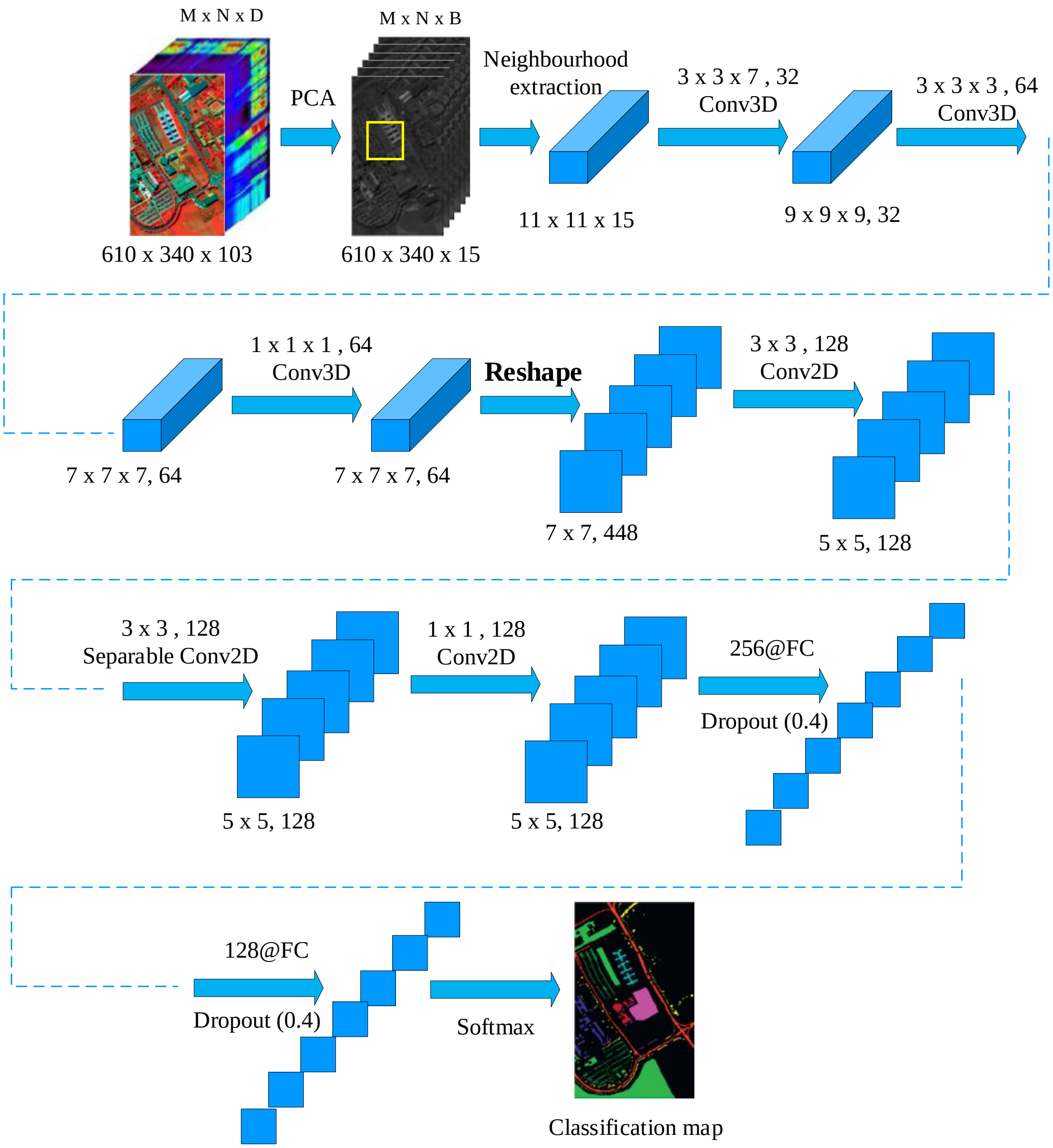

Fig. 3. Proposed hybrid 3D/2D CNN method for HSIC

In Equations (12), (13) and (14), K is the number of classes used for HSIC. M represents the confusion matrix. The confusion matrix is created by comparing the predicted classification map with ground truth that is actual value. M_{ii} shows the main diagonal values of the confusion matrix and is the number of correctly classified samples belonging to class i. N is the total number of test samples. M_{i+} indicates the total number of samples in row i, while M_{+i} indicates the total number of samples in column i.

The proposed method (PM) has been compared with HSIC methods based on deep learning from the literature, such as SVM-RBF [41], 2D CNN [42], 3D CNN [21], SSRN [32], and HybridSN[27].

In the first application, IP dataset is used. Classification accuracy of each class according to class-based and OA, AA and K evaluation metrics is shown in Table III. Ground truth and the classification maps obtained as a result of the applications performed with the IP dataset are given in Fig. 4. When Table III is examined, it is seen that the proposed method obtained the best classification result with 99.32%, 99.46% and 99.22%, respectively, according to the OA, AA and K evaluation metrics. The closest results to the proposed method were obtained with SSRN with 99.19% OA, 98.93% AA, and 99.07% K values, HybridSN with 99.26% OA, 98.98% AA, and 99.09% K. Among the other classification methods, the lowest OA, AA and K values were obtained in SVM-RBF with 82.83%, 80.84% and 82.23%, respectively. Considering the AA value in SVM-RBF, it is seen that the number of correct classifications for each class is lower than other methods. With the proposed method in class-based classification, the best classification accuracies of 100% are obtained in all classes except for the 2nd, 3rd, 5th, 10th, 11th, 12th and 16th classes. The best classification accuracy for class 2 was obtained in the SSRN method with a difference of 0.57% from the proposed method. The best classification accuracy for class 3 was obtained in the SSRN method with a difference of 1.79% from the proposed method. Similarly, the best classification accuracy was obtained in SSRN with 99.24% for class 5, HybridSN with 98.76% for class 10, PM with 99.75% for class 11, HybridSN with 99.12% for class 12, and PM with 98.65% for class 16.

TABLE III
NUMBER OF TRAINING AND TEST SAMPLES OF IP DATASET AND CLASSICATION ACCURACIES (%)

No. of classes	Train/Test	SVM-RBF	2D CNN	3D CNN	SSRN	HybridSN	PM
1	9/37	61.5	85.88	94.63	97.82	97.98	100
2	286/1142	78.68	91.31	93.9	99.17	98.37	98.6
3	166/664	73.41	91.07	94.85	99.53	99.48	97.74
4	47/190	71.58	80.38	93.48	97.79	97.38	100
5	97/386	80.38	91.89	93.56	99.24	99.23	98.96
6	146/584	92.27	99.01	94.2	99.51	99.14	100
7	6/22	79.52	82.59	89.73	98.7	99	100
8	96/382	87.38	100	96.01	99.85	100	100
9	4/16	85.87	66.55	95	98.5	99.01	100
10	194/778	77.58	86.38	94.55	98.74	98.76	98.71
11	491/1964	83.75	90.47	93.87	99.3	99.64	99.75
12	119/474	83.21	82.89	91.52	98.43	99.12	98.95
13	41/164	84.64	99.06	93.89	100	100	100
14	253/1012	98.01	97.86	91.77	99.31	100	100
15	77/309	94.3	90.52	95.03	99.2	99.35	100
16	19/74	61.43	98.94	93.57	97.82	97.26	98.65
OA (%)		82.83	90.89	94.07	99.19	99.26	99.32
AA (%)		80.84	89.68	93.72	98.93	98.98	99.46
Kappa x 100		82.23	88.56	93.87	99.07	99.09	99.22

The second application is performed with the PU dataset. The classification accuracies obtained as a result of the applications performed with the PU dataset are given in Table IV. The classification maps obtained as a result of the classification are shown in Fig. 5. When Table IV is examined, it is seen that the proposed method obtained the best classification result with 99.83%, 99.70% and 99.78%, respectively, according to the OA, AA and K evaluation metrics. The proposed method obtained 0.11%, 0.1%, 0.14% better results in terms of OA, AA and K values, respectively, than HybridSN, which is one of the most advanced methods. In addition, the proposed method obtained 0.21%, 0.21% and 0.28% better results in terms of OA, AA and K values, respectively, compared to SSRN, which is another of the most advanced methods. The lowest classification results are obtained in the SVM-RBF method with 82.67% OA, 80.84% AA and 81.21% K values. Classification accuracies of 96.89% OA, 95.79% AA, 2D CNN with 96.56% K and 3D CNN with 99.07% OA, 98.75% AA, 98.87% K were obtained, respectively. When class-based classification accuracies are compared, the proposed method has the best classification performance with 100% in the 5th, 6th and 9th classes. In addition, it is seen that the best classification results are obtained with 99.95% in class 1, 99.98% in class 2, 99.53% in class 4 and 99.46% in class 8. The best classification results are obtained in the HybridSN method, with 99.01% in class 3 and 99.69% in class 7. Considering all classification accuracies, it is seen that the classification map obtained by the proposed method is quite close to the ground truth.

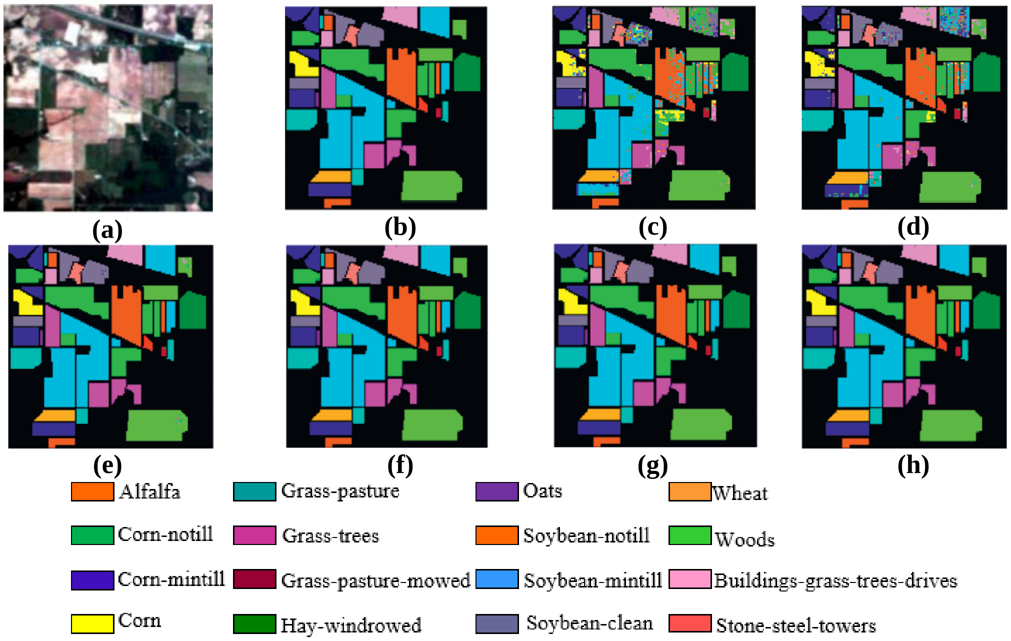

Fig. 4. Classification maps obtained as a result of predicted for IP. (a) False-color image, (b) ground truth, (c) SVM-RBF, (d) 2D CNN, (e) 3D CNN, (f) SSRN, (g) HybridSN, and (h) Proposed method

TABLE IV
NUMBER OF TRAINING AND TEST SAMPLES OF PU DATASET AND CLASSICATION ACCURACIES (%)

No. of classes	Train/Test	SVM-RBF	2D CNN	3D CNN	SSRN	HybridSN	PM
1	663/5968	93.68	97.37	97.4	99.75	99.76	99.95
2	1865/16784	97.02	99.26	94.73	99.79	99.78	99.98
3	210/1889	82.41	80.73	95.05	98.29	99.01	98.94
4	306/2758	96.51	95.54	98.04	99.52	99.53	99.53
5	135/1211	98.38	99.75	99.01	99.82	99.87	100
6	503/4526	90.01	93.14	98.62	99.77	99.77	100
7	133/1197	85.92	91.65	97.02	99.65	99.69	99.42
8	368/3314	88.08	92.39	98.23	99.05	99.21	99.46
9	95/852	99.85	99.09	99.29	99.78	99.80	100
OA (%)		82.67	96.89	99.07	99.62	99.72	99.83
AA (%)		80.84	95.79	98.75	99.49	99.60	99.70
Kappa x 100		81.21	96.56	98.87	99.50	99.64	99.78

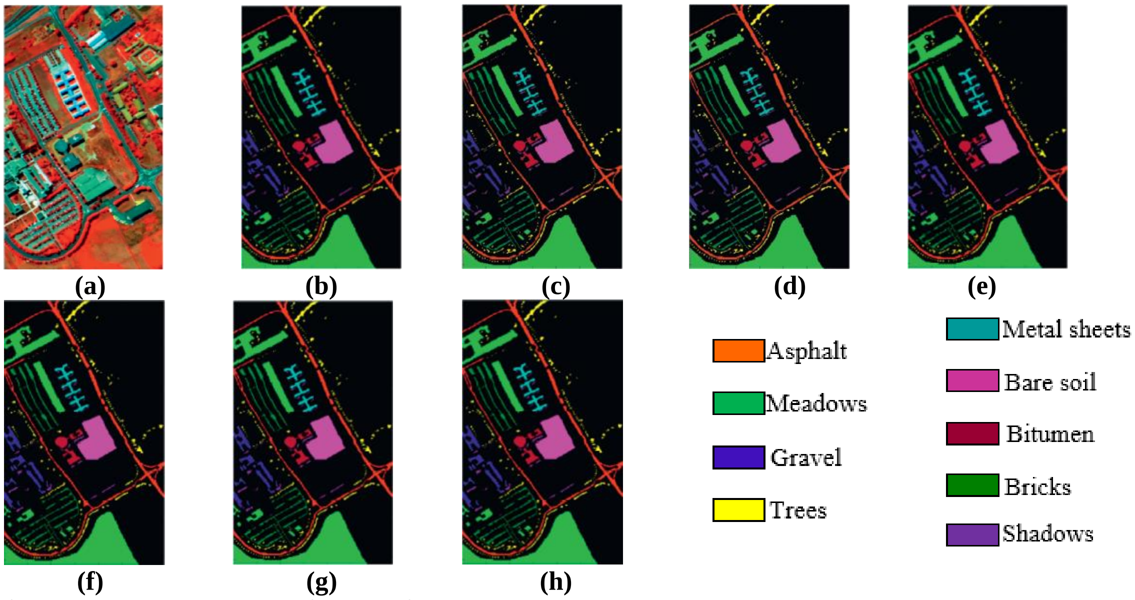

Fig. 5. Classification maps obtained as a result of predicted for PU. (a) False-color image, (b) ground truth, (c) SVM-RBF, (d) 2D CNN, (e) 3D CNN, (f) SSRN, (g) HybridSN, and (h) Proposed method

The third application is performed with the SA dataset. The classification accuracies obtained as a result of the applications performed with the SA dataset are given in Table V. The classification maps obtained as a result of the classification are shown in Fig. 6. When Table V is examined, the best classification accuracies were obtained with 99.90% OA, 99.93% AA and 99.89 K values in the proposed method. The closest results to the proposed method were obtained in the HybridSN method with 99.80% and SSRN method with 99.64% OA, 98.76% AA, 99.60% K values in all evaluation metrics. The proposed method achieved better classification accuracy of 7.23% OA, 8.32% AA and 7.68% K, respectively, from SVM-RBF, 4.56% OA, 5.18% AA and 4.96% K from 2D CNN, and 5.88% OA, 6.44% AA and 6.32% K from 3D CNN. When the class-based classification results were examined, the proposed method achieved the best classification accuracy with 100% in the 1st, 2nd, 3rd, 6th, 7th, 9th, 11th, 12th, 13th, 16th classes. The best classification results were obtained in the HybridSN method with 100% for the 4th, 5th, 8th, 10th and 14th classes. The proposed method with 99.63% for class 15 gives the best results. Considering Table V and Figure 6, the classification maps obtained by the proposed method and HybridSN methods are very similar to the ground truth.

TABLE V
NUMBER OF TRAINING AND TEST SAMPLES OF SA DATASET AND CLASSICATION ACCURACIES (%)

No. of classes	Train/Test	SVM-RBF	2D CNN	3D CNN	SSRN	HybridSN	PM
1	201/1808	92.56	95.51	95.19	99.78	100	100
2	373/3353	94.78	95.87	96.12	99.78	100	100
3	198/1778	94.22	95.64	95.89	99.78	100	100
4	139/1255	97.36	99.99	97.18	99.99	100	99.76
5	268/2410	93.38	94.51	94.70	99.78	100	99.92
6	396/3563	95.36	96.71	95.63	99.78	100	100
7	358/3221	95.79	95.51	95.36	99.78	100	100
8	1127/10144	80.21	86.93	86.33	99.78	100	99.85
9	620/5583	97.67	97.42	97.19	99.78	100	100
10	328/2950	88.99	90.93	90.94	92.69	100	99.90
11	107/961	90.20	94.24	90.20	90.68	98.24	100
12	193/1734	96.42	99.91	96.24	99.95	99.61	100
13	92/824	95.53	95.53	95.31	99.85	99.59	100
14	107/963	91.26	92.97	94.18	99.88	100	99.90
15	727/6541	72.67	90.93	81.39	99.12	99.32	99.63
16	181/1626	89.37	93.43	94.09	99.78	100	100
OA (%)		92.67	95.34	94.02	99.64	99.80	99.90
AA (%)		91.61	94.75	93.49	98.76	99.80	99.93
Kappa x 100		92.21	94.93	93.57	99.60	99.80	99.89

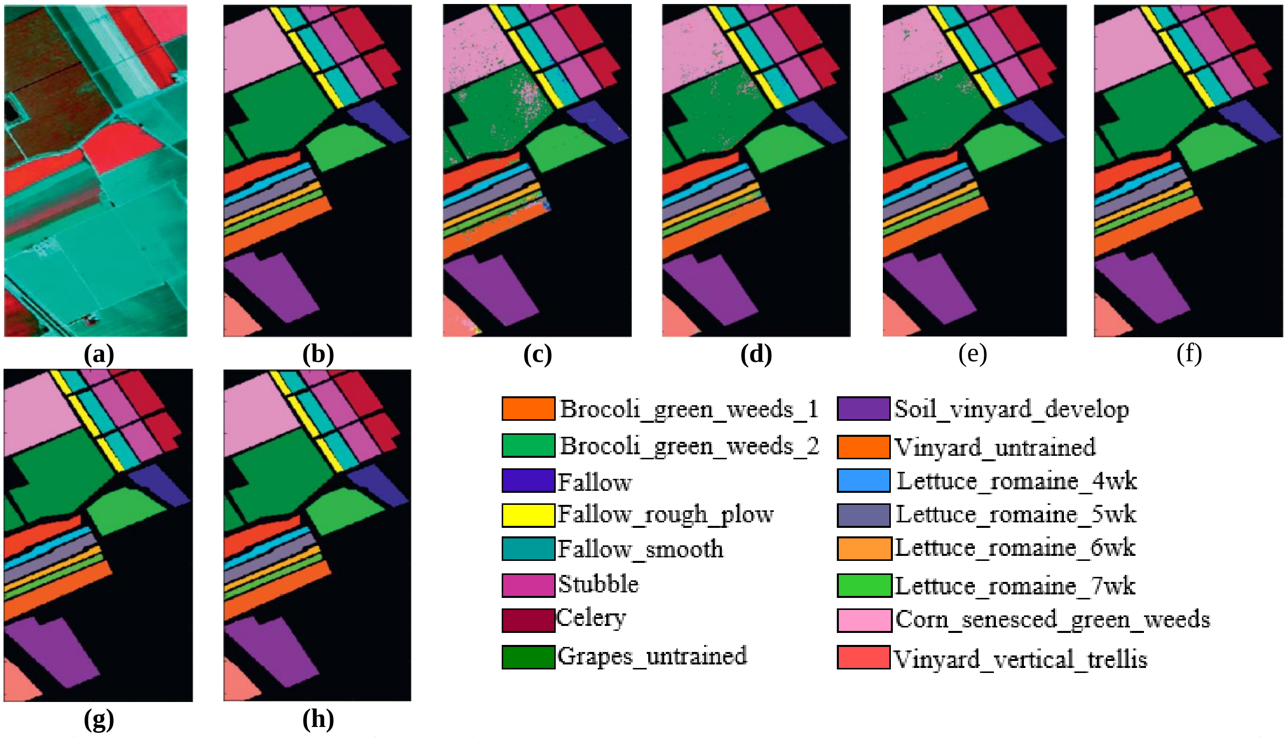

Fig. 6. Classification maps obtained as a result of predicted for SA. (a) False-color image, (b) ground truth, (c) SVM-RBF, (d) 2D CNN, (e) 3D CNN, (f) SSRN, (g) HybridSN, and (h) Proposed method

IV. Conclusion

In this study, a hybrid depthwise separable CNN method for HSIC is proposed. The method combines 3D/2D CNN and spatially depthwise separable convolution layers. While spatial-spectral features are extracted with 3D CNN, spatial features are continuously extracted using 2D CNN. With depthwise separable convolution, the number of trainable parameters and computation time are reduced and more spatial features are extracted. Considering the applications performed with IP, PU and SA benchmark datasets, with the proposed method, 99.32% OA, 99.46% AA, 99.22% K in IP, 99.83% OA, 99.70% AA, 99.78% K in PU, 99.90% OA, 99.93% AA, 99.89% K values in SA were obtained. In line with all these values, the proposed method was compared with SVM-RBF, 2D CNN, 3D CNN, SSRN, HybridSN methods. When Table III-IV-V is examined, it is seen that better classification performance is achieved with the proposed method. In future studies, Dilated convolution, which is another convolution method, will be examined first to solve the HSIC problem. Later, applications will be developed on how it affects HSIC performance when used with depthwise separable convolution.

References

[1] H. Gao, Y. Yang, C. Li, L. Gao, and B. Zhang, "Multiscale Residual Network with Mixed Depthwise Convolution for Hyperspectral Image Classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 4, pp. 3396–3408, 2021, doi: 10.1109/TGRS.2020.3008286.

[2] H. Fırat and D. Hanbay, "4CF-Net: Hiperspektral uzaktan algılama görüntülerinin spektral uzamsal sınıflandırılması için yeni 3B evrişimli sinir ağı," Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, vol. 1, pp. 439–453, 2021, doi: 10.17341/gazimmfd.901291.

[3] H. Firat, M. Uçan, and D. Hanbay, "Classification of Hyperspectral Remote Sensing Images Using Hybrid 3D-2D CNN Architecture," Journal of Computer Science, vol. IDAP-2021, no. Special, pp. 132–140, 2021.

[4] Y. Wang, W. Yu, and Z. Fang, "Multiple Kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information," Remote Sensing, vol. 12, no. 1, 2020, doi: 10.3390/RS12010120.

[5] M. Ahmad et al., "Spatial prior fuzziness pool-based interactive classification of hyperspectral images," Remote Sensing, vol. 11, no. 9, pp. 1–19, 2019, doi: 10.3390/rs11091136.

[6] A. Alcolea, M. E. Paoletti, J. M. Haut, J. Resano, and A. Plaza, "Inference in supervised spectral classifiers for on-board hyperspectral imaging: An overview," Remote Sensing, vol. 12, no. 3, pp. 1–29, 2020, doi: 10.3390/rs12030534.

[7] J. S. 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, doi: 10.1109/TGRS.2004.842481.

[8] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq, "Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 7570–7588, 2021, doi: 10.1109/JSTARS.2021.3099118.

[9] M. F. Özdemir and D. Hanbay, "Comparison of Optimization Algorithms for Multi-Object Tracking FairMOT Algorithm," Journal of Computer Science, vol. IDAP-2021, no. Special, pp. 147–153, 2021.

[10] İ. Sel and D. Hanbay, "Creating a Parallel Corpora for Turkish-English Academic Translations," Journal of Computer Science, vol. IDAP-2021, no. Special, pp. 335–340, 2021.

[11] G. MENDİ and C. BUDAK, "Automatic Cell Nucleus Segmentation Using Superpixels and Clustering Methods in Histopathological Images," Balkan Journal of Electrical and Computer Engineering, vol. 9, no. 3, pp. 304–309, 2021, doi: 10.17694/bajece.864266.

[12] C. Budak, V. Mençik, and V. Gider, "Determining similarities of COVID-19 – lung cancer drugs and affinity binding mode analysis by graph neural network-based GEFA method," Journal of Biomolecular Structure and Dynamics, vol. 0, no. 0, pp. 1–13, 2021, doi: 10.1080/07391102.2021.2010601.

[13] H. Üzen, H. Fırat, A. Karcİ, and D. Hanbay, "Automatic Thresholding Method Developed With Entropy For Fabric Defect Detection," in 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019, pp. 14–17.

[14] H. Uzen, M. Turkoglu, and D. Hanbay, "Texture defect classification with multiple pooling and filter ensemble based on deep neural network," Expert Systems with Applications, vol. 175, no. March, p. 114838, 2021, doi: 10.1016/j.eswa.2021.114838.

[15] C. Zhao, X. Wan, G. Zhao, B. Cui, W. Liu, and B. Qi, "Spectral-Spatial Classification of Hyperspectral Imagery Based on Stacked Sparse Autoencoder and Random Forest," European Journal of Remote Sensing, vol. 50, no. 1, pp. 47–63, 2017, doi: 10.1080/22797254.2017.1274566.

[16] H. Data et al., "Deep Learning-Based Classification of Hyperspectral Data," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 6, pp. 2094–2107, 2014, doi: 10.1109/JSTARS.2014.2329330.

[17] A. Mughees and L. Tao, "Efficient deep auto-encoder learning for the classification of hyperspectral images," Proc. - 2016 Int. Conf. Virtual Real. Vis. ICVRV 2016, no. September, pp. 44–51, 2017, doi: 10.1109/ICVRV.2016.16.

[18] P. Zhong, Z. Gong, S. Li, and C. B. Schonlieb, "Learning to Diversify Deep Belief Networks for Hyperspectral Image Classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 6, pp. 3516–3530, 2017, doi: 10.1109/TGRS.2017.2675902.

[19] Y. Chen, X. Zhao, and X. Jia, "Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381–2392, 2015, doi: 10.1109/JSTARS.2015.2388577.

[20] J. Li, B. Xi, Y. Li, Q. Du, and K. Wang, "Hyperspectral classification based on texture feature enhancement and deep belief networks," Remote Sensing, vol. 10, no. 3, 2018, doi: 10.3390/rs10030396.

[21] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," Remote Sensing, vol. 9, no. 1, 2017, doi: 10.3390/rs9010067.

[22] C. BUDAK, V. MENÇİK, and M. E. ASKER, "Effect on model performance of regularization methods," DÜMF Mühendislik Dergisi, vol. 5, pp. 757–765, 2021, doi: 10.24012/dumf.1051352.

[23] H. S. Nogay, T. C. Akinci, and M. Yilmaz, "Detection of invisible cracks in ceramic materials using by pre-trained deep convolutional neural network," Neural Computings and Applications, vol. 0123456789, 2021, doi: 10.1007/s00521-021-06652-w.

[24] C. Zhang et al., "Joint Deep Learning for land cover and land use classification," Remote Sensing of Environment, vol. 221, no. May 2018, pp. 173–187, 2019, doi: 10.1016/j.rse.2018.11.014.

[25] H. Firat, M. E. Asker, and D. Hanbay, "Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D CNN," Remote Sensing Applications: Society and Environment, p. 100694, 2022, doi: 10.1016/j.rsase.2022.100694.

[26] P. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232–6251, 2016, doi: 10.1049/iet-ipr.2019.1282.

[27] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3D-2D CNN Feature Hierarchy for Hyperspectral Image Classification," arXiv, vol. 17, no. 2, pp. 277–281, 2019.

[28] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A Fast and Compact 3-D CNN for Hyperspectral Image Classification," IEEE Geoscience and Remote Sensing Letters, no. April, pp. 1–5, 2020, doi: 10.1109/LGRS.2020.3043710.

[29] Z. Ge, G. Cao, X. Li, and P. Fu, "Hyperspectral Image Classification

Method Based on 2D-3D CNN and Multibranch Feature Fusion," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 5776–5788, 2020, doi: 10.1109/JSTARS.2020.3024841.

[30] H. C. Mingyi He, Bo Li, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," 2017 IEEE International Conference on Image Processing (ICIP), pp. 3904–3908, 2017.

[31] C. Mu, Z. Guo, and Y. Liu, "A multi-scale and multi-level spectral-spatial feature fusion network for hyperspectral image classification," Remote Sensing, vol. 12, no. 1, 2020, doi: 10.3390/RS12010125.

[32] 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, doi: 10.1109/TGRS.2017.2755542.

[33] A. Mohan and M. Venkatesan, "HybridCNN based hyperspectral image classification using multiscale spatiospectral features," Infrared Physics and Technology, vol. 108, no. March, 2020, doi: 10.1016/j.infrared.2020.103326.

[34] A. Mohan and V. Meenakshi Sundaram, "V3O2: hybrid deep learning model for hyperspectral image classification using vanilla-3D and octave-2D convolution," Journal of Real-Time Image Processing, no. 0123456789, 2020, doi: 10.1007/s11554-020-00966-z.

[35] F. Cao and W. Guo, "Deep hybrid dilated residual networks for hyperspectral image classification," Neurocomputing, vol. 384, pp. 170–181, 2020, doi: 10.1016/j.neucom.2019.11.092.

[36] 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, doi: 10.1109/TGRS.2018.2818945.

[37] M. Ahmad, S. Shabbir, R. A. Raza, M. Mazzara, S. Distefano, and A. M. Khan, "Hyperspectral Image Classification: Artifacts of Dimension Reduction on Hybrid CNN," no. January, pp. 1–9, 2021.

[38] L. Jiang, B. Zhu, and Y. Tao, "Hyperspectral Image Classification Methods," Hyperspectral Imaging for Food Quality Analysis and Control, pp. 79–98, 2010, doi: 10.1016/B978-0-12-374753-2.10003-6.

[39] A. Wang, C. Liu, D. Xue, H. Wu, Y. Zhang, and M. Liu, "Depthwise separable relation network for small sample hyperspectral image classification," Symmetry (Basel)., vol. 13, no. 9, 2021, doi: 10.3390/sym13091673.

[40] L. Dang, P. Pang, and J. Lee, "Depth-wise separable convolution neural network with residual connection for hyperspectral image classification," Remote Sensing, vol. 12, no. 20, pp. 1–20, 2020, doi: 10.3390/rs12203408.

[41] B. C. Kuo, H. H. Ho, C. H. Li, C. C. Hung, and J. S. Taur, "A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 1, pp. 317–326, 2014, doi: 10.1109/JSTARS.2013.2262926.

[42] Q. Wang, J. Gao, and Y. Yuan, "A Joint Convolutional Neural Networks and Context Transfer for Street Scenes Labeling," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 5, pp. 1457–1470, 2018, doi: 10.1109/TITS.2017.2726546.

BIOGRAPHIES

HÜSEYİN FIRAT received the B.S. degree in computer engineering from Cukurova University, Adana, Turkey, in 2014. He received the M.S. degree in computer engineering from Inonu University in 2018. He is currently working toward the Ph.D. degree in the computer engineering. He also works as a lecturer at Dicle University in Turkey. His current interests include remote sensing, deep learning, and hyperspectral image classificaiton.

MEHMET EMİN ASKER was born in Diyarbakır, Turkey, in 1976. He received the B.S. degree in electrical electronics engineering, from Firat University, Elazig, Turkey in 1993, the M.S. degree and the Ph.D. degree in electrical machines, power electronics from Firat University, Elazig, Turkey, in 2009 and 2016, respectively. He is an Assistant Professor with Dicle University, Department of electrical power and energy. Where he teaches courses on power system, power electronics, circuit theory and electrical machines since 2007. His research interests include electrical machines, power electronics and power systems.

DAVUT HANBAY received the B.S., M.S. and Ph.D. degreesin electric–electronic engineering from Firat University,Elazig, Turkey, in 1999, 2003, and 2007, respectively. He is the author or coauthor of more than 30 papers in journals and conferences. His current interests include classification, optimization and pattern recognition.