IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 3 December 2022, accepted 20 December 2022, date of publication 21 December 2022, date of current version 28 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3231579

RESEARCH ARTICLE

Pixel-Wise Classification of Hyperspectral Images With 1D Convolutional SVM Networks

MAYAR A. SHAFAEY[1], FARID MELGANI[2], (Fellow, IEEE), MOHAMMED A.-M. SALEM[3], MARYAM N. AL-BERRY[4], HALA M. EBIED[4], EL-SAYED A. EL-DAHSHAN[1,5], AND MOHAMED F. TOLBA[4], (Senior Member, IEEE)

[1]Faculty of Computer and Information Technology, National Egyptian E-Learning University, Cairo 12611, Egypt
[2]Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy
[3]Faculty of Media Engineering and Technology, German University in Cairo, Cairo 11835, Egypt
[4]Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
[5]Faculty of Science, Ain Shams University, Cairo 11566, Egypt

Corresponding author: Mayar A. Shafaey (maliattia@eelu.edu.eg)

This work was supported in part by the Department of Information Engineering and Computer Science, University of Trento, Italy.

ABSTRACT Nowadays, remote sensing image analysis is needed in various important tasks such as city planning, land-use classification, agriculture monitoring, military surveillance, and many other applications. In this context, hyperspectral images can play a useful role, but require specific handling. This paper presents a convolutional neural network based on one-dimensional support vector machine (SVM) convolution operations (1D-CSVM) for the analysis of hyperspectral images. SVM-based CNN (CSVM) was introduced first for the classification of high spatial resolution RGB images. It relies on linear SVMs to create filter banks in the convolution layers. In this work, the network is modified to cope with one-dimensional hyperspectral signatures and perform pixel-based classification. It thus analyzes each pixel spectrum independently from the pixel spatial neighborhood. Experiments were carried out on four benchmark hyperspectral datasets, Salinas-A, Kennedy Space Center (KSC), Indian Pines (IP) and Pavia University (Pavia-U). Compared to state-of-the-art models, the proposed network produces promising results for all tested datasets, with an accuracy up to 99.76%.

INDEX TERMS Convolutional neural network, feedforward learning, hyperspectral signature, machine learning, pixel-based classification, support vector machine.

I. INTRODUCTION

The satellite image is an image of the whole or part of the earth taken using artificial satellites. It can either be visible light images, water vapor images or infrared images [1]. The different types of satellites produce (high spatial, radiometric, temporal, and spectral) resolution images that cover the whole Earth in less than a day [2], [3], [4], [5]. The large-scale nature of these data sets introduces new challenges in image analysis. Indeed, the analysis and classification of those remote sensing images have been considered as hot topics recently.

In fact, the hyperspectral images are composed of hundreds spectral bands for the same scene. It has an important feature which aids in differentiating materials of interest. Actually, it has detailed spectral information which raises substantially the power of discrimination [8].

In this work, we aim to exploit the full potential of the spectral information conveyed by each image pixel by merging the convolutional support vector machine (CSVM) [9], [10], which is an alternative supervised learning strategy based on support vector machines (SVMs), and (1D-CNN) [11] approach in one network. We call the proposed architecture one dimensional convolutional support vector machine (1D-CSVM).

Basically, it processes and analyzes the spectral signature of each pixel through a cascade of multiple convolutional

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Di Martino.

TABLE 1. Survey of recent publications based on deep learning methods for hyperspectral images (HSI) classification.

Reference	Year	Purpose	Datasets	Strategy	Results
[14]	2015	HSI classification in spectral domain	Several hyperspectral images	Deep CNN	Better performance than conventional classification methods
[15]	2016	HSI Classification	3 public benchmark HSI	2D-CNN	Outperforms other state-of-art methods
[16]	2016	Objects classification based on image pixels	HSI	CNN	Significant performance
[17]	2016	Spectral-spatial information for hyperspectral classification	3 benchmark hyperspectral datasets	Five-layers CNN	~97.5 %
[18]	2017	Hyperspectral Image Classification	Indian Pines	CNN with adaptive convolutional kernels	97.84 ± 0.2249
[19]	2017	Spectral-Spatial HSI classification	Indian Pines, and Pavia Scene	3D-CNN	Increasing the accuracy compared to traditional ANN techniques
[20]	2018	HSI target detection	4 hyperspectral images	1D-CNN	Outperforms classical target detection algorithms
[21]	2018	Spectral-Spatial hyperspectral image classification	3 hyperspectral images	Deep learning	Competitive results
[22]	2018	HSI classification	KSC, SA, Pavia-U, and IP	Hybrid 2D/3D-CNN	99.28%, 98.97%, 99.57%, and 99.09%
[23]	2018	Spectral-spatial classification of HSI	5 groups of hyperspectral images	CNN with a single hidden layer	Higher than 98 %
[9]	2018	Analysis of 2D Remote Sensing (RS) imagery	2 UAV datasets (vehicles and solar panel)	Convolutional Support Vector Machine (CSVM)	~ 97%
[24]	2019	Reliable HSI classification	3 public benchmark HSI	CNN and deep residual network ensemble	> 95 %
[25]	2020	Automatic detection of oil palm trees	UAV images	Region-based CNN	97.8 %
[10]	2020	Very High Resolution (VHR) RS image analysis	3 VHR and 2 UAV datasets	CSVM	Competitive results
[6]	2021	HIS classification with few labeled samples	KSC, Salinas scene, and Pavia-U,	CNN	94.2%, 97.83%, and 96.10%
[7]	2021	HSI classification with spatial consistence	IP, and Salinas scene	Fully convolutional spatial propagation network	99.6 %
[38]	2022	Accurate classification of HSI with limited labeled samples	KSC, IP, Houston U, and Salinas scene	Compressed synergic deep convolution neural network with Aquila optimization (CSDCNN-AO)	94.44 %
[39]	2022	HSI classification	IP, and Salinas scene	Enhancing-CNN (e-CNN)	95.81%, and 98.39%

and reduction layers and ends by a classification layer. Each convolutional layer in 1D-CSVM uses the linear SVMs as filter banks to generate a set of feature vectors to reduce the number of trainable parameters and improve the speed of data processing. Indeed, spectral information is very important to decide the nature of each point on the ground. In this work, we aim at exploiting as much as possible the high potential of this rich information source.

II. RELATED WORK

From a classification perspective, a common paradigm to analyze hyperspectral data is the pixel-based approach, in which the single image pixels are classified by means of the spectral information they convey. Another approach consists to exploit spectral-spatial features [12]. Recently, convolutional neural networks (CNNs) [13] have shown particularly effective for several analysis tasks including segmentation, classification, and object detection. CNNs either require a large amount of training data or have to be fine-tuned on the specific dataset and thus classification task. Table 1 reports some of the recent works which applied CNNs for various hyperspectral scene classification tasks and provided promising outcomes.

III. PROPOSED NETWORK ARCHITECTURE

Basically, and like traditional CNNs, 1D-CSVM consists of an input layer, several convolutional and reduction layers followed by a classification layer, see Fig.1.

A. INPUT LAYER

Firstly, the traditional convolutional network takes the whole image as an input for the first and initial layer. Here, in the proposed network, all image pixels are stored in a matrix data structure as an input. That matrix has $m \times n$ dimension

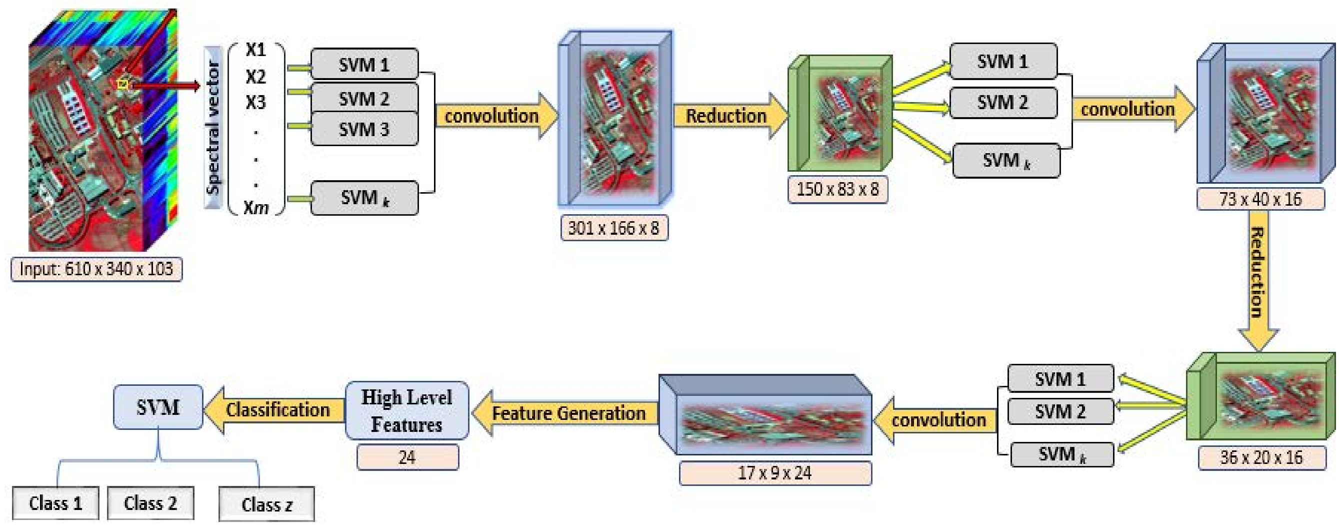


FIGURE 1. 1D-CSVM model with 3 convolutional layers, 2 reduction layers, and 1 classification layer at the top of the network. We suppose the number of SVM filters are 8, 16, and 24 with kernel size 9 × 9, 5 × 5, and 3 × 3 for convolutional layers 1, 2, and 3, respectively.

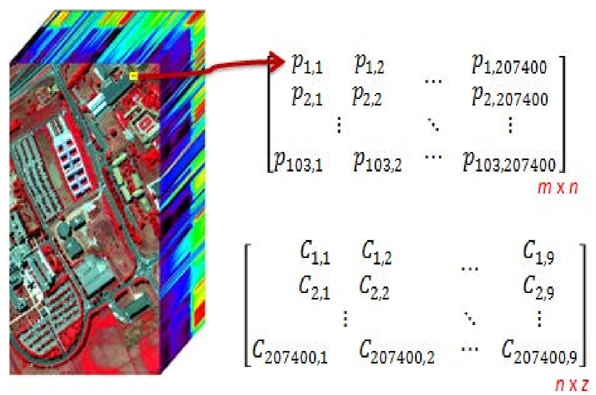


FIGURE 2. Pavia-U (left), its input matrix (upper right), and its corresponding label matrix of groundtruth classes (lower right).

size, where m represents the total number of image spectrum bands, whereas n represents the total number of image pixels. Moreover, since learning is supervised, each input matrix has a corresponding output (label) matrix which has $n \times z$ dimension size, where z indicates the total number of ground truth classes, see Fig.2.

Fig. 2 represents the hyperspectral image, Pavia-U. This image has a 610×340 dimension size, 103 spectrum bands, and its ground truth image has 9 different classes. Hence, we can represent this hyperspectral image as an input matrix with 103×207400 dimension size. And its corresponding label matrix with size 207400×9.

B. CONVOLUTIONAL LAYER

For feature vector generation, each convolutional layer in the network convolves the feature vectors provided by the previous layer with SVM filter banks. Firstly, the original input spectrum is convolved with SVM filters. The produced feature vector is then convolved again by the latter layers. So, the three main steps for the convolution process are:

1) TRAINING SET CONSTRUCTION

Each pixel of the hyperspectral image is considered as an individual vector; thus, we can obtain a global training set as previously represented in the first input layer by its class label.

2) SVM FILTERS GENERATION

The SVM weights are generated directly for each convolutional layer in a supervised and feedforward manner, unlike in conventional CNNs where weights are estimated via back-propagation. For each SVM filter, the weight vector w and bias b are computed by the following unconstrained optimization equation:

$$min_{w,b} w^T w + C \sum_{i=1}^{l} \mathcal{E}(w, b; x_{i_i}, y_i) \quad (1)$$

where C is a penalty parameter, $\mathcal{E}\ (w, b; x_{i_i}, y_i)$ is a loss function [26].

SVM filters are trained on different sub-training sets $\{x_{i_i}, y_i\}^{l_{i=1}}$ of size l, which are randomly sampled from the global training set. The number of SVM filters k for each convolution layer is determined empirically by different trials. Each filter will represent different features of a class, see Fig.3. Then, the complete weights of these SVM filters are grouped into one filter bank whose outcome is then flattened into a 1D vector to be ready for the next step.

3) FEATURE VECTOR GENERATION

In the convolutional step, each pixel vector is convolved with the k different SVM filters, for extracting both nonlinear features from the considered pixel, which are combined in 1D vector, for generating a hyper-feature vector, H. Moreover,

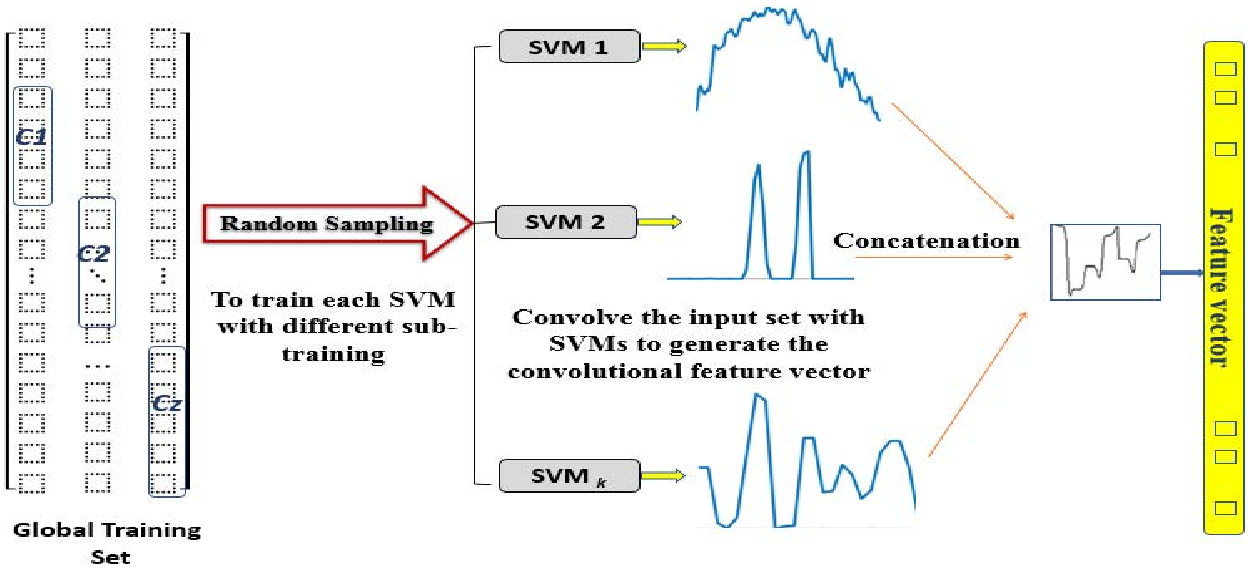


FIGURE 3. Feature vector obtained by 1st SVM convolutional layer. Each SVM filter represents different features of the spectral bands. These feature vectors are then concatenated into one vector to feed it for the next convolutional layer.

this feature vector is then passed to a Rectified Linear Unit (ReLU), a nonlinear gating function, for keeping only the positive values. The next convolutional layers do the same operations. Only the first layer is taking the input image, and the latter takes the input as the feature vector produced by the precedent layer as shown in Fig. 1.

C. REDUCTION LAYER

As in the conventional CNN, the reduction/pooling layer in 1D-CSVM works in a similar way. It subsamples small blocks from the convolutional layer to produce a single output from each block. For reducing the size of the representation, this layer is placed between two successive convolutional layers.

D. CLASSIFICATION LAYER

After multiple SVM convolutional and reduction layers, high-level features are obtained and fed to the last layer, which is usually known as the classification (or prediction) layer. It uses a linear SVM classifier again to classify the high-level representations obtained by the network. It is trained on the hyper-feature vectors extracted from the last layer to produce a final prediction (see Fig. 1). In our experiments, the multi-class SVM is implemented using the one-versus-all method.

IV. EXPERIMENTAL VALIDATION

A. DATASETS

In the experiments, we assessed the proposed 1D-CSVM on four benchmark hyperspectral images namely Salinas-A, Kennedy Space Center (KSC), Indian Pines (IP), and Pavia University (Pavia-U) [32].

Firstly, Salinas-A is a sub-scene which acquired from Salinas Valley, USA. It is captured by the AVIRIS sensor [$\sim$ 3.7 m/pixel spatial resolution]. It covers the area which includes 86 lines by 83 samples. This scene includes bare soils, vegetables, and vineyard grounds. It includes only six classes and background, as shown in Fig. 4. Table 2 -A (columns: 2 and 3) indicates the ground truth classes for the Salinas-A scene and their respective samples number.

Secondly, the KSC data set was obtained by 18-m spectrometer over Florida, USA. That scene has a size of 512 $\times$ 614 $\times$ 176, and contains 13 classes as ground truth, see Fig. 5 and Table 2 -B (columns: 2 and 3).

Thirdly, The University of Pavia scene was captured in 2003 from a flight over Northern Italy (Pavia) by the ROSIS sensor ($\sim$1.3-m/ pixel). Its dimensions are 610 $\times$ 340 $\times$ 103, and it has 9 ground truth classes which cover the ground as shown in Fig. 6 and Table 2 -C (columns: 2 and 3).

Finally, the scene in the test site 'Indian pines' was collected over North-western Indiana. It is captured by the 224-band AVIRIS sensor, by the 0.4–2.5 10-6 m wavelength range. It composes of 145 $\times$ 145 pixels. It covers 16 classes of agricultural, forest, and road areas, as shown in Fig. 7. Table 2 -D (columns: 2 and 3) indicates the classes of the ground truth for the Indian pines scene.

B. EXPERIMENTAL DESIGN

In this article, we will focus on the Salinas-A dataset to illustrate our model. Initially, we present the results of our experiments using a three-layer 1D-CSVM. Table 3 shows the parameters of each layer in the network including convolutional, reduction, and classification layers.

First, for the input layer, we can represent Salinas-A dataset as a matrix of 224 $\times$ 7138 dimension size; where 7138 represents the total number of image pixels and 224 represents the total number of image spectrum bands. Also, the label matrix has a 7138 $\times$ 6, where 6 is the total number

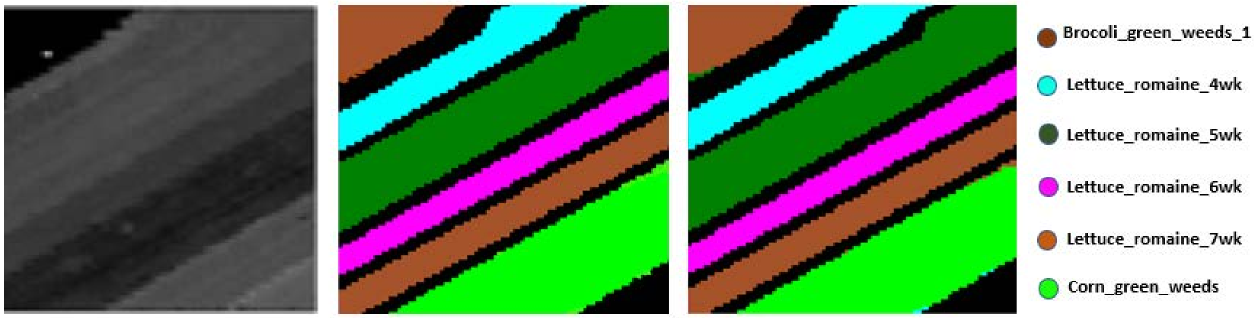


FIGURE 4. The original input (left), the ground truth data (middle), and the classification map (right) for Salinas-A.

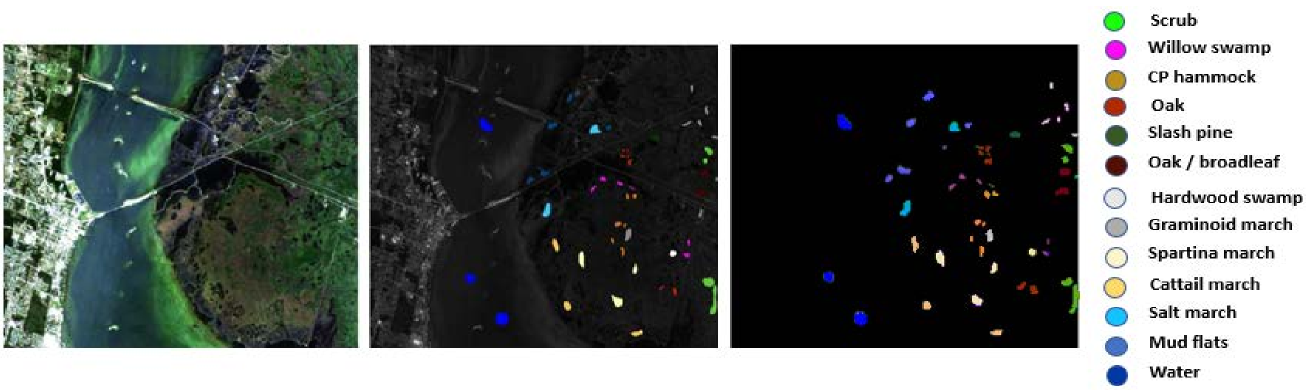


FIGURE 5. The original input (left), the ground truth data (middle), and the classification map (right) for KSC.

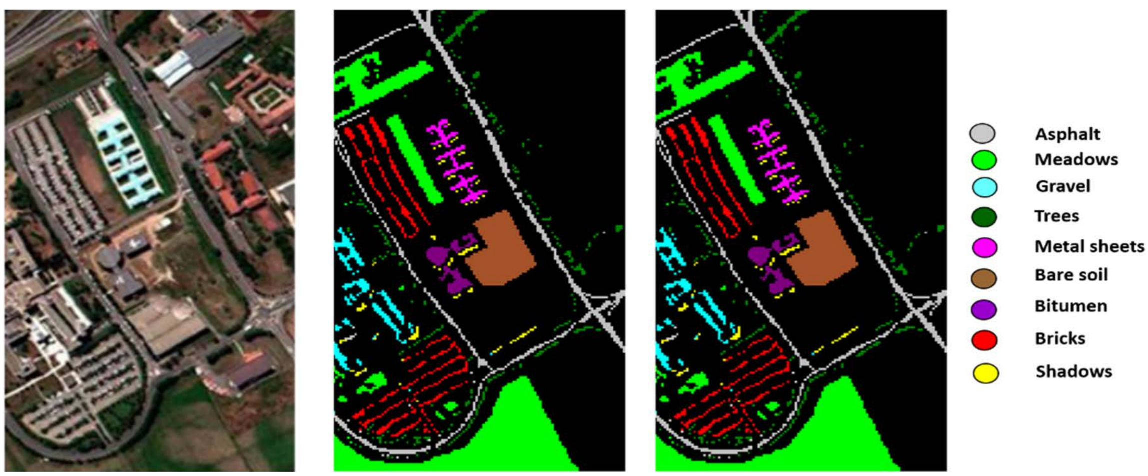


FIGURE 6. The original input (left), the ground truth data (middle), and the classification map (right) for Pavia-U.

of ground-truth classes. We excluded the background ∼1790 pixels and partitioned the foreground pixels randomly into 30:70, 50:50, or 70:30 for training and validation sets.

Second, for the convolutional layer, we trained each SVM filter by extracting randomly l samples from the training dataset. For layers 1, 2, and 3, $l = 49$, 25, and 9, respectively. These values were chosen through several trials. To compute the weights of the SVM filters, a *Liblinear-multicore-2.11-1* software package is used [27]. The penalty parameter C is estimated through a threefold cross-validation method for each SVM. It is normally ranging from 10^{-1} to 10^{3}. After that, the convolution operation is done using *vl_nnvonv* of MatConvNet, introduced by *A. Vedaldi* and *K. Lenc* [28].

Third, for the reduction layer, the 'max_pooling' operator is used for the three layers created. Each reduction layer has a different window size and stride value (for more details, refer

FIGURE 7. The original input (left), the ground truth data (middle), and the classification map (right) for IP.

to Table 3). Finally, the last layer in the network produces the high-level representations to feed it again into a linear SVM classifier for carrying out the classification task.

To evaluate the performance, the results are presented in terms of Average Accuracy (AA), Overall Accuracy (OA), and Kappa coefficient (K), see Table 2. 'AA' is defined as the average of the accuracy values measured over each class, while the 'OA' is the ratio between total number of correct classifications and total number of samples in the test data, and 'K' is a statistical metric through qualitative items [29]. The accuracy value of each class is computed based on the average of Producer's Accuracy (PA) and User's Accuracy (UA), see the following equations as in (2) and (3), shown at the bottom of the page.

For the other datasets, KSC, Pavia-U, and IP the network parameters are the same as in the table above, except for the 'SVM_window_size' and 'sample_train_size' parameters. The SVM window size (kernel size) of layers 1, 2, and 3 are 9×9, 5×5, and 3×3, respectively. The number of training samples is 81, 25, and 9 for the three layers, respectively. For each dataset, the accuracy results are averaged by running five different trials (random training samples).

V. RESULTS AND DISCUSSIONS

The following figures provide information about the feature vectors formed via several convolutional layers and reduced by the pooling layers of 1D-CSVM and trained on Salinas-A dataset. Fig. 8 illustrates the mean value over 224 spectral bands for the 6 classes.

After applying the convolution and reduction operations of each layer of the 3-layer network, the number of features is reduced for each pixel vector. It reached 111 (Fig. 9-Top), 57 (Fig. 9-Middle), and 27 (Fig. 9-Bottom), respectively. So, each pixel vector started with 224 features

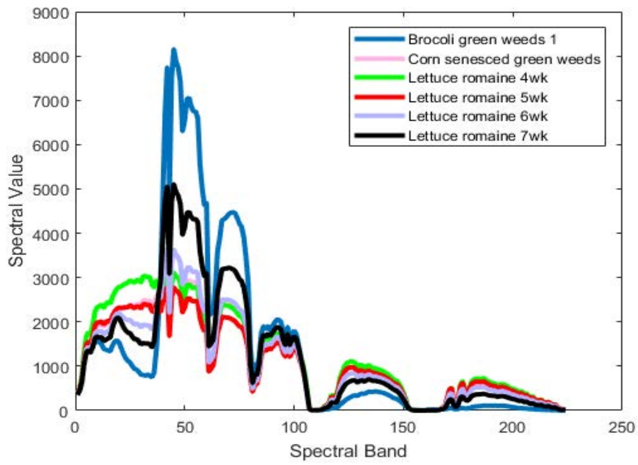


FIGURE 8. Mean spectral signature for Salinas-A.

and ended by a 27-dimensional feature vector fed to the SVM classifier.

In order to study the effect of the different filter sizes on the network performances, we have computed the training accuracy percentage of our different experiments on Salinas-A, KSC, Pavia-U, and IP datasets, as shown in Table 4 (A).

The main four factors that may affect our network performance are 1) number of network layers; 2) number of SVM filters in each convolutional layer; 3) window size of SVM kernels; and 4) training-to-testing ratio. In general, not all these factors significantly affect the overall performance accuracy. For example, the three-network architecture is enlarged by one additional convolutional and reduction layer. Then, the classification accuracy has enhanced by a small fractional number, but, on the other hand, it consumed double the execution time. Also, the number of SVM kernels

$$PA = \frac{\text{correctly identified pixels}}{\text{total number of pixels/class}} \tag{2}$$

$$UA = \frac{\text{correctly identified pixels}}{\text{correctly identified pixels} + \text{incorrectly identified pixels}} \tag{3}$$

TABLE 2. A) Classification results of each class for *Salinas-A* scene B) Classification results of each class for *KSC* scene C) Classification results of each class for *Pavia*-U D) Classification results of each class for *IP* scen.

A)

Class #	Class Name	# pixels	1D-CSVM Method
1	Brocoli_green_weeds_1	391	100 %
2	Corn_green_weeds	1343	100 %
3	Lettuce_roamine_4wk	616	99 %
4	Lettuce_roamine_5wk	1525	99 %
5	Lettuce_roamine_6wk	674	100 %
6	Lettuce_roamine_7wk	799	100 %
OA			**99.66 %**
AA			**99.66 %**
K			**98.28 %**

B)

Class #	Class Name	# pixels	1D-CSVM Method
1	Spartina march	520	95 %
2	Cattail march	404	98 %
3	CP hammock	256	99.4 %
4	Slash Pine	252	97 %
5	Oak	161	99 %
6	Hardwood	229	98 %
7	Mud flats	503	99 %
8	water	927	98.2 %
9	Swamp	105	98 %
10	Graminoid march	390	99 %
11	Scrub	347	99.2 %
12	Willow swamp	243	98 %
13	Salt march	419	100 %
OA			**98.03 %**
AA			**98.29 %**
K			**97.22 %**

C)

Class #	Class Name	# pixels	1D-CSVM Method
1	Bare soil	5029	99.9 %
2	Bitumen	1330	100 %
3	Trees	3064	100 %
4	Shadows	947	99.9 %
5	Painted metal sheets	1345	100 %
6	Meadows	18649	99.9 %
7	Asphalt	6631	100 %
8	Self-blocking bricks	3682	100 %
9	Gravel	2099	100 %
OA			**99.76 %**
AA			**99.96 %**
K			**98.48 %**

D)

Class #	Class Name	# pixels	1D-CSVM Method
1	Oats	20	100 %
2	Stone-Steel-Towers	93	97 %
3	Wheat	205	92.2 %
4	Soybean-clean	593	97.4 %
5	Buildings-Grass-Trees-Drives	386	95.14 %
6	Soybean-mintill	2455	93 %
7	Hay-windrowed	478	100 %
8	Woods	1265	100 %
9	Alfalfa	46	100 %
10	Soybean-notill	972	90.5 %
11	Grass-trees	730	95 %
12	Corn-notill	1428	94.2 %
13	Grass-pasture	483	98.3 %
14	Grass-pasture-mowed	28	95.15 %
15	Corn	237	96.3 %
16	Corn-mintill	830	100 %
OA			**96.20 %**
AA			**96.51 %**
K			**96.46 %**

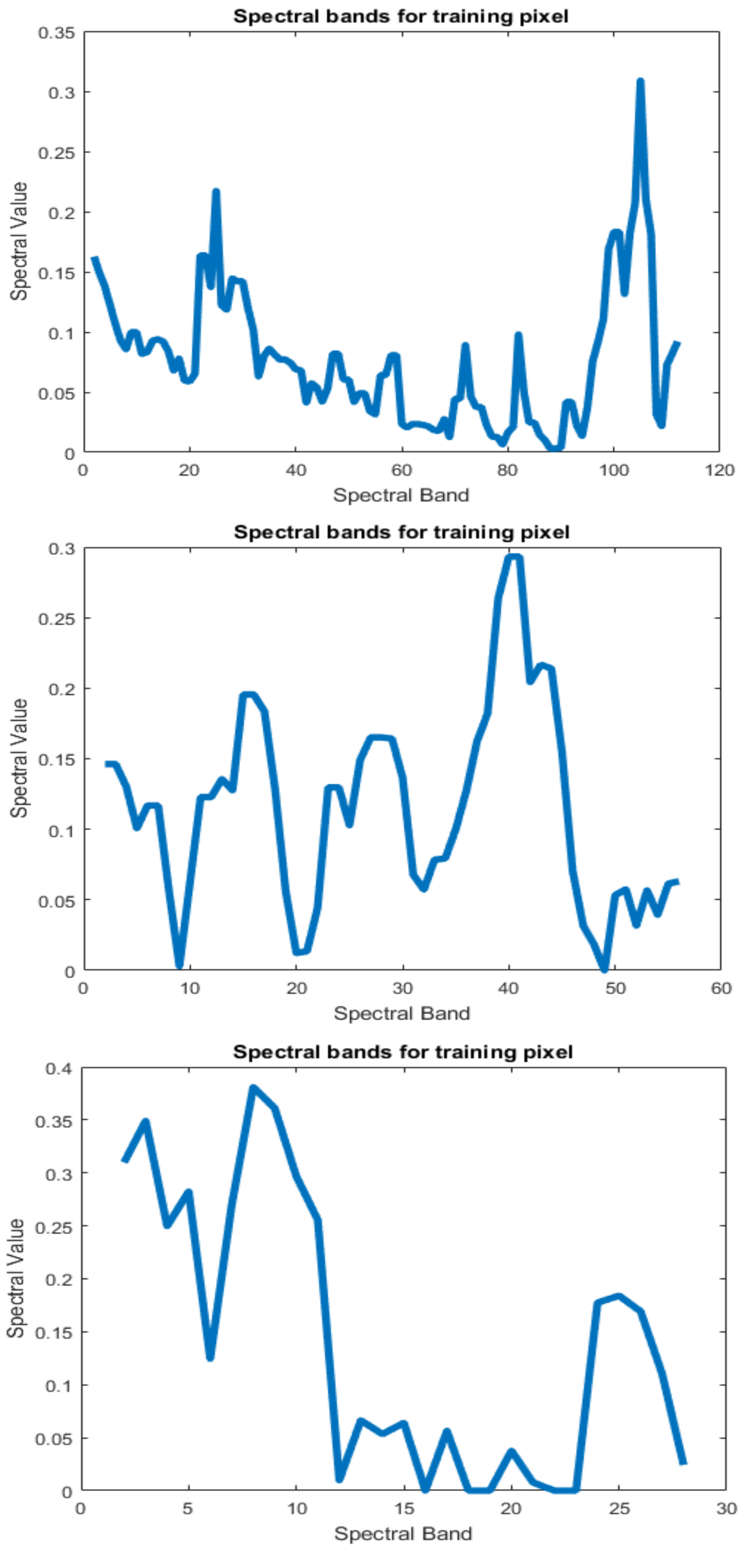


FIGURE 9. Example of feature vector representation by layers (1-3) for one-pixel vector of Salinas-A.

in each convolutional layer improved the overall accuracy slightly, as shown in Table 4 (A). It is firstly set to a small number and then incremented until a suitable accuracy value is reached through several trials.

TABLE 3. **Parameters defined for each layer of 1d-csvm network for *salinas-A*.**

Parameter	Layer 1	Layer 2	Layer 3
SVM_window_size	7 × 7	3 × 3	3 × 3
pooling_window_size	3 × 3	3 × 3	2 × 2
convolutional_stride	2	2	2
convolutional_pad	0	0	0
pooling_pad	0	0	0
pooling_stride	2	2	2
pooling_type	max	max	max
# SVM_filters	8	16	24
sample_train_size (*l*)	49	25	9

However, the most critical parameter is the size of the SVM window. Hence, the experiments are carried out on 5 different sizes of the three layers: (3 × 3, 3 × 3, 3 × 3), (5 × 5, 5 × 5, 3 × 3), (7 × 7, 3 × 3, 3 × 3), (9 × 9, 5 × 5, 3 × 3), and (11 × 11, 7 × 7, 5 × 5). A large mask size may reduce dramatically the classification accuracy. Finally, the training-to-testing ratio; as known, in most experiments, when the training set is increased, the classification accuracy is enhanced too. We run the experiments on several training-testing ratios (30:70, 50:50, and 70:30). The best window size credits to dimension (7 × 7, 3 × 3, 3 × 3) for Salinas-A, and (9 × 9, 5 × 5, 3 × 3) for KSC, Pavia-U, and IP (refer previous section and Table 3).

It is worth taking into consideration that the network corresponding to this configuration using SVM filter banks in the convolution operation is significantly reducing the training computation time, in comparison with conventional CNN. This enhancement is due to the feedforward strategy in SVM weights computation, instead of the backpropagation approach. This allows to add a new layer without re-building the network again.

TABLE 4. **A) Effect of SVM filters size of the 1d-csvm convolution layers on the classification accuracy through 70:30 training: validation ratio B) Effect of several training: validation ratios on the overall classification accuracy, average, accuracy and kappa coefficient.**

A)

Window size for 3 layers		# SVM filters for three layers						# SVM filters for three layers			# SVM filters for three layers			# SVM filters for three layers	
		2, 4, 8	**4, 8, 12**	**8, 8, 16**	**12, 16, 24**	**8, 16, 24**		**12, 16, 24**	**8, 16, 24**		**12, 16, 24**	**8, 16, 24**		**12, 16, 24**	**8, 16, 24**
(3x3, 3x3, 3x3)	**SALINAS-A**	90.4%	90.56%	90.8%	91%	92.33%	**KSC**			**PAVIA-U**			**INDIAN PINES**		
(5x5, 5x5, 3x3)		95%	95.3%	95.7%	96.01%	96.3%									
(7x7, 3x3, 3x3)		97.2%	97.5%	98.2%	98.59%	**99.66%**		98.19%	97.88%		89.59%	90.08%		92.60%	94.18%
(9x9, 5x5, 3x3)		93.5%	93.2%	93.8%	94.12%	94.19%		98.02%	**98.03%**		91.76%	**99.76%**		93.48%	**96.20%**
(11x11, 7x7, 5x5)		88%	88.6%	90.4%	90.3%	91.7%		94.89%	95.37%		88.3%	88.78%		90.53%	91%

B)

Training: validation ratio	# SVM filters for three layers (8, 16, 24) with window size (7×7, 3×3, 3×3) for Salinas-A, and (9×9, 5×5, 3×3) for KSC, Pavia-U, and IP														
		AA (%)	OA (%)	K (%)		AA (%)	OA (%)	K (%)		AA (%)	OA (%)	K (%)		AA (%)	OA (%)	K (%)
30: 70	**SALINAS-A**	96.50	96.50	96.15	**KSC**	95.37	95.33	92.45	**PAVIA-U**	97.80	97.80	98.00	**INDIAN PINES**	93.2	92.91	92.68
50: 50		97.64	97.64	97.71		96.68	96.68	95.14		98.06	97.23	96.87		94.74	94.73	93.44
70: 30		99.66	99.66	98.28		98.03	97.87	97.22		99.76	99.03	98.48		96.20	96.15	96.46

TABLE 5. **Comparison between the proposed 1D-CSVM and state-of-art in terms of AA for *Salinas-A*, *KSC*, *Pavia-U*, and *IP* datasets.**

Reference(s)	Model	Tr: Ts ratio	*Salinas-A*	*KSC*	*Pavia-U*	*IP*	# Trainable Parameters
[29] & [31]	*ResNet*	Pre-trained models	98.09%	93.23%	98.40%	99.03%	Millions of learnable parameters
[33] & [34]	*AlexNet*		97.22%	96.30%	92.62%	88.26%	
[34] & [35]	RBF-SVM	70: 30	91.66%	97.25%	90.52%	87.60%	-
[34]	2D- CNN	70: 30	98.90%	97.81%	98.75%	96.37%	~ 172, 007
[36] & [37]	1D- CNN	90: 10	95.47%	91.80%	93.50%	83.40%	-
[29]	Lightweight 3D-CNN	Transfer learning	-	97.87%	99.68%	98.87%	~ 763, 008
[38]	(CSDCNN-AO)	-	94.44%	93.44%	-	94.44%	-
[40]	Deep 3-DCNN	70:30	98.87%	-	-	96.28%	233, 680
[41]	SP-CNN	Transfer learning	96.12%	-	93.40%	93.57%	~ 1, 600, 000
Proposed Method	**1D-CSVM**	**70: 30**	**99.66%**	**98.03%**	**99.76%**	**96.20%**	**~64, 504**
		Processing Time (minutes)	**(0.3 min)**	**(4.7 min)**	**(4.12 min)**	**(1.34 min)**	

The results we achieved are compared to the state-of-the-art, especially, against the pre-trained CNNs [2]. The training time of CNN models is typically very long, even using high capability GPUs. They have a large number of convolutional and reduction layers. For instance, classic and inception models consist of more than 20 consecutive layers. As well, residual models which are known as ultra-deep models may consist of more than 50 layers. They require for instance from 5 to 6 days and from 2 to 3 weeks for *AlexNet* and *ResNet* training, respectively. Also, those models require millions of training data i.e. *ImageNet* [30] for image classification usage.

The comparison between the created 1D-CSVM and state-of-art is provided in Table 5 on the aforementioned datasets. By comparing our proposed network versus the pre-trained models, the proposed model provides better results. It shows an interesting behavior in a pixel-wise classification scenario like the one considered in this paper.

Moreover, all our experiments require modest machines, i.e. hardware equipped by NVIDIA GTX 1050 4G with compute capability 6.1 GPU: Intel® Core™i7-7700HQ @ 2.20GHz, and 16 GB RAM. As well, the carried-out experiments take few minutes in the training process; as the proposed network has only three convolutional layers, two reduction layers, and one classification layer. All of these properties of 1D-CSVM exhibit clearly a better behavior in time and accuracy for testing large-scale datasets. The output classification maps for the test data are shown in the Fig. 4 - Fig. 7.

VI. CONCLUSION

In this work, a novel 1D-CSVM network is proposed for the pixel-based classification of hyperspectral images. This network is mainly based on state-of-the-art SVM. It exploits it as filters for feature vector generation of each convolutional process, and for classifying the high-level features. Moreover, the proposed network has important properties: 1) to compute SVM filter weights, it uses a feedforward supervised learning strategy. 2) it consists of few (in these experiments just three) convolutional layers, in contrast with the Deep CNN structures. 3) it does not require a large number of training samples. 4) it consumes few hours in training compared to pre-trained CNNs. The results obtained over four known hyperspectral datasets, Salinas-A, KSC, Pavia-U, and IP confirmed its effectiveness in terms of accuracy and execution time compared to the methods in the literature. In the future, it can be extended to handle other datasets for model generalization. Moreover, a 3D version (Spectral-Spatial feature-based network) could be envisioned to capture spatial information as well in the convolution and analysis process.

REFERENCES

[1] L. G. Z. Xia, T. Wu, L. Lin, and X. Tai, "Deep learning for remote sensing image understanding," *J. Sensors*, vol. 2016, pp. 1–2, Jan. 2016.

[2] M. A. Shafaey, M. A.-M. Salem, H. M. Ebeid, M. N. Al-Berry, and M. F. Tolba, "Comparison of CNNs for remote sensing scene classification," in *Proc. 13th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2018, pp. 27–32.

[3] K. Ose, T. Corpetti, and L. Demagistri, "Multispectral satellite image processing," in *Optical Remote Sensing of Land Surface*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 57–124.

[4] M. Claverie, V. Demarez, B. Duchemin, O. Hagolle, D. Ducrot, C. Marais-Sicre, J.-F. Dejoux, M. Huc, P. Keravec, P. Béziat, R. Fieuzal, E. Ceschia, and G. Dedieu, "Maize and sunflower biomass estimation in Southwest France using high spatial and temporal resolution remote sensing data," *Remote Sens. Environ.*, vol. 124, pp. 844–857, Sep. 2012.

[5] M. Herold, M. E. Gardner, and D. A. Roberts, "Spectral resolution requirements for mapping urban areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1907–1919, Sep. 2003.

[6] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.

[7] Y. Jiang, Y. Li, S. Zou, H. Zhang, and Y. Bai, "Hyperspectral image classification with spatial consistence using fully convolutional spatial propagation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10425–10437, Jan. 2021.

[8] D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, Jan. 2002.

[9] Y. Bazi and F. Melgani, "Convolutional SVM networks for object detection in UAV imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3107–3118, Jun. 2018.

[10] Y. Li, F. Melgani, and B. He, "CSVM architectures for pixel-wise object detection in high-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6059–6070, Sep. 2020.

[11] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[12] E. Li, A. Samat, W. Liu, C. Lin, and X. Bai, "High-resolution imagery classification based on different levels of information," *Remote Sens.*, vol. 11, no. 24, p. 2916, Dec. 2019.

[13] M. A. Shafaey, M. A.-M. Salem, H. M. Ebied, M. N. Al-Berry, and M. F. Tolba, "Deep learning for satellite image classification," in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2018, pp. 383–391.

[14] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.

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

[16] M. Längkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and segmentation of satellite orthoimagery using convolutional neural networks," *Remote Sens.*, vol. 8, no. 4, p. 329, Apr. 2016.

[17] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li, "Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 5067–5070.

[18] C. Ding, Y. Li, Y. Xia, W. Wei, L. Zhang, and Y. Zhang, "Convolutional neural networks based hyperspectral image classification method with adaptive kernels," *Remote Sens.*, vol. 9, no. 618, pp. 1–15, 2017.

[19] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogram. Remote Sens.*, vol. 145, pp. 120–147, Nov. 2018.

[20] J. Du and Z. Li, "A hyperspectral target detection framework with subtraction pixel pair features," *IEEE Access*, vol. 6, pp. 45562–45577, 2018.

[21] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, and A. De Giorgi, "New frontiers in spectral–spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

[22] Y. Luo, J. Zou, C. Yao, X. Zhao, T. Li, and G. Bai, "HSI-CNN: A novel convolution neural network for hyperspectral image," in *Proc. Int. Conf. Audio, Lang. Image Process. (ICALIP)*, Jul. 2018, pp. 464–469.

[23] J. Acquarelli, E. Marchiori, L. Buydens, T. Tran, and T. Laarhoven, "Spectral–spatial classification of hyperspectral images: Three tricks and a new learning setting," *Remote Sens.*, vol. 10, no. 7, p. 1156, Jul. 2018.

[24] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.
[25] X. Liu, K. H. Ghazali, F. Han, and I. I. Mohamed, "Automatic detection of oil palm tree from UAV images based on the deep learning method," *Appl. Artif. Intell.*, vol. 35, no. 1, pp. 13–24, Jan. 2021.
[26] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale L2-loss linear support vector machines," *J. Mach. Learn. Res.*, vol. 9, pp. 1369–1398, Jun. 2008.
[27] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[28] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 689–692.
[29] H. Zhang, Y. Li, Y. Jiang, P. Wang, Q. Shen, and C. Shen, "Hyperspectral classification based on light weight 3D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.
[30] J. Deng, W. Dong, R. Socher, Li-Jia Li, K. Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.
[31] Z.-Y. Wang, Q.-M. Xia, J.-W. Yan, S.-Q. Xuan, J.-H. Su, and C.-F. Yang, "Hyperspectral image classification based on spectral and spatial information using multi-scale ResNet," *Appl. Sci.*, vol. 9, no. 22, p. 4890, Nov. 2019.
[32] *Salinas—A, KSC, IP, and Pavia-U Datasets*. Accessed: Dec. 22, 2022. [Online]. Available: https://rslab.ut.ac.ir/data
[33] C. Li, S. X. Yang, Y. Yang, H. Gao, J. Zhao, X. Qu, Y. Wang, D. Yao, and J. Gao, "Hyperspectral remote sensing image classification based on maximum overlap pooling convolutional neural network," *Sensors*, vol. 18, no. 10, p. 3587, Oct. 2018.
[34] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Apr. 2018.
[35] H. Hasan, H. Z. M. Shafri, and M. Habshi, "A comparison between support vector machine (SVM) and convolutional neural network (CNN) models for hyperspectral image classification," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 357, no. 1, Nov. 2019, Art. no. 012035.
[36] T.-H. Hsieh and J.-F. Kiang, "Comparison of CNN algorithms on hyperspectral image classification in agricultural lands," *Sensors*, vol. 20, no. 6, p. 1734, Mar. 2020.
[37] J. Nalepa, M. Antoniak, M. Myller, P. R. Lorenzo, and M. Marcinkiewicz, "Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation," *Microprocessors Microsyst.*, vol. 73, Mar. 2020, Art. no. 102994.
[38] T. S. Reddy, J. Harikiran, M. K. Enduri, K. Hajarathaiah, S. Almakdi, M. Alshehri, Q. N. Naveed, and M. H. Rahman, "Hyperspectral image classification with optimized compressed synergic deep convolution neural network with Aquila optimization," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–14, Jul. 2022.
[39] M. Kavitha, R. Gayathri, K. Polat, A. Alhudhaif, and F. Alenezi, "Performance evaluation of deep e-CNN with integrated spatial–spectral features in hyperspectral image classification," *Measurement*, vol. 191, Mar. 2022, Art. no. 110760.
[40] M. Kavitha and R. Gayathri, "Joint spectral-spatial feature using deep 3-D CNN for hyperspectral images," in *Proc. Int. Conf. Electron. Syst. Intell. Comput. (ICESIC)*, Apr. 2022, pp. 281–285.
[41] F. Xie, Q. Gao, C. Jin, and F. Zhao, "Hyperspectral image classification based on superpixel pooling convolutional neural network with transfer learning," *Remote Sens.*, vol. 13, no. 5, p. 930, Mar. 2021.

MAYAR A. SHAFAEY was born in Cairo, Egypt, in 1991. She received the B.S. and M.S. degrees in scientific computing from Ain Shams University, Cairo, in 2012 and 2017, respectively, where she is currently pursuing the Ph.D. degree with the Faculty of Computers and Information Science.

From December 2018 to March 2019, she was a short-term Researcher with the Doctoral School in Information and Communication Technology (ICT), University of Trento, Italy. She is also working as an Assistant Lecturer with the Faculty of Computer and Information Technology, National Egyptian E-Learning. Her research interests include remote sensing, image processing, machine learning, and deep learning. She has published more than ten articles in these areas.

FARID MELGANI (Fellow, IEEE) received the degree in electronics (state engineering) from the University of Batna, Batna, Algeria, in 1994, the M.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1999, and the Ph.D. degree in electronic and computer engineering from the University of Genoa, Genoa, Italy, in 2003.

From 1999 to 2002, he cooperated with the Signal Processing and Telecommunications Group, Department of Biophysical and Electronic Engineering, University of Genoa. Since 2002, he has been an Assistant Professor and then an Associate Professor in telecommunications with the University of Trento, Trento, Italy, where he has taught pattern recognition, machine learning, radar remote sensing systems, and digital transmission. He is currently the Head of the Signal Processing and Recognition Laboratory, Full Professor, and the Dean of Undergrad and Grad Studies, Department of Information Engineering and Computer Science, University of Trento. He has coauthored over 190 scientific publications. His research interests include remote sensing, signal/image processing, pattern recognition, machine learning, and computer vision.

Dr. Melgani has served on the scientific committees of several international conferences. He is also an Associate Editor of the IEEE Geoscience and Remote Sensing Letters, the *International Journal of Remote Sensing*, and *Remote Sensing*. He is currently an Associate Editor of the IEEE Transactions on Geoscience and Remote Sensing and IEEE Journal on Miniaturization for Air and Space Systems. He is a referee for numerous international journals.

MOHAMMED A.-M. SALEM was born in Cairo, Egypt. He received the B.S. degree in the field of mathematics and computer science, in 1997, the M.Sc. degree in scientific computing from Ain Shams University, Cairo, in 2002, and the Ph.D. degree in multiresolution image segmentation from Humboldt-Universität in Berlin, Germany, in November 2008.

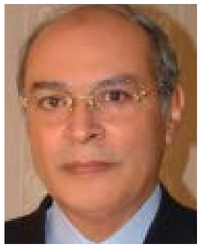
From 2009 to 2016, he was an Assistant Professor at the Faculty of Computer and Information Sciences, Ain Shams University. Currently, he is a Professor in image and computer vision at the Faculty of Media Engineering and Technology, German University in Cairo. He has published a book and over than 100 papers in the fields of image/video segmentation and content-based retrieval. He has gained teaching experience and supervised different graduation projects during his work at Ain-Shams University and Humboldt-Universität of Berlin. He was an Internal Researcher at the Knowledge Media Institute (KMI), Milton Keynes, U.K., in Summer 2008. He was a short-term Postdoctoral Researcher at the Heinz Nixdorf Institute, Paderborn, Germany, in Summer 2009 and Summer 2010, and Humboldt-Universität of Berlin, in Summer 2012 and Summer 2014. He was also an Internal Researcher at the University of Trento, Italy. His research interests include 3D vision, multimedia analytics, remote sensing, computer vision, signal processing, machine learning, multiresolution analysis, and wavelet transform.

MARYAM N. AL-BERRY was born in Cairo, Egypt. She received the B.S., M.Sc., and Ph.D. degrees in scientific computing from Ain Shams University, in 2001, 2007, and 2015, respectively. From 2020 to 2021, she was worked as the Coordinator of three credit hour programs, namely, digital multimedia, artificial intelligence, and cyber security. In 2021 and 2022, she became the Coordinator of Digital Multimedia Program. She is currently the Coordinator of Bioinformatics Program. She is also an Assistant Professor at the Scientific Computing Department, Faculty of Computers and Information Sciences, Ain Shams University. Her research interests include image processing and analysis, action and activity recognition, satellite image segmentation, machine learning, and deep learning.

HALA M. EBIED received the B.Sc. degree in pure mathematics and computer science from the Faculty of Science, Ain Shams University (ASU), Egypt, the M.Sc. and Ph.D. degrees in computer and information sciences from ASU, in 2002 and 2009, respectively, and the Ph.D. degree from the Heinz Nixdorf Institute, University of Paderborn, Germany, in 2008.

Since December 2016, she has been working as the Director of the Quality Unit, Faculty of Computer and Information Sciences (FCIS), ASU, where she has also been working as the Head of the Scientific Computing Department, FCIS, since January 2018. Since August 2018, she has also been working as the Vice Dean of Education and Students Affairs with FCIS, ASU, where she has also been a Professor with FCIS, since 2019. She has coauthored about 70 research articles in refereed journals and international conferences. Her research interests include neural networks and deep learning, computer vision, machine learning, and robotics.

EL-SAYED A. EL-DAHSHAN has been a Professor in scientific computer at Ain Shams University, Cairo, Egypt, since 2012. He has been a Professor in computational physics, since 2018. His research interests include time-frequency representation and machine learning and deep learning techniques and their applications in the fields of signal and image processing, optimization techniques based on meta-heuristic approaches, implementation of digital signal, and image processing real time systems on the FPGA chips. His list of publications includes more than 100 articles in intelligent computing approaches in physics, medical signals and images, and remote sensing images analysis.



MOHAMED F. TOLBA (Senior Member, IEEE) has been a Professor in scientific computing with Ain Shams University, since 1984, where he was the Vice President from 2002 to 2006 and the Dean of the Faculty of Computers and Information Sciences from 1996 to 2002. He has supervised more than 90 M.Sc. and 50 Ph.D. degrees at Ain Shams University and other Egyptian universities. He is currently a Consultant to different local and international organizations for IT. He has more than 220 publications in the fields of AI, image processing, pattern recognition, OCR, scientific computing, and simulation and modeling.

He was a member of the International Association for Science and Technology for Development (IASTED), Canada, from 1995 to 2007, the International Society for Computers and their Applications (ISCA), USA, from 1998 to 2007, the Advisory Committee of Strengthening Science and Technology Researchers Project—STRP Ministry of Scientific Research, from 2006 to 2009, and the Committee for Evaluation of Egyptian Space Program of the National Authority for Remote Sensing and Space Sciences—Ministry of Scientific Research. He has been a member of the Association for Computing Machinery (ACM), USA, since 2000, the Software Engineering Competence Center (SECC), since 2004, the Information Technology Academic Collaboration (ITAC), since 2005, and the E-Learning Committee Board, since 2008. He is also the honorary chairperson of many international conferences and the chairperson of several IT sector committees in Egypt.