Reduced-cost hyperspectral convolutional neural networks

Giorgio Morales,[a] John W. Sheppard,[a,*] Bryan Scherrer,[b] and Joseph A. Shaw[b]

[a]Montana State University, Gianforte School of Computing, Bozeman, Montana, United States
[b]Montana State University, Department of Electrical and Computer Engineering, Bozeman, Montana, United States

Abstract. Hyperspectral imaging provides a useful tool for extracting complex information when visual spectral bands are not enough to solve certain tasks. However, processing hyperspectral images (HSIs) is usually computationally expensive due to the great amount of both spatial and spectral data they incorporate. We present a low-cost convolutional neural network designed for HSI classification. Its architecture consists of two parts: a series of densely connected three-dimensional (3-D) convolutions used as a feature extractor, and a series of two-dimensional (2-D) separable convolutions used as a spatial encoder. We show that this design involves fewer trainable parameters compared to other approaches, yet without detriment to its performance. What is more, we achieve comparable state-of-the-art results testing our architecture on four public remote sensing datasets: Indian Pines, Pavia University, Salinas, and EuroSAT; and a dataset of Kochia leaves [*Bassia scoparia*] with three different levels of herbicide resistance. The source code and datasets are available online. (Hyper3DNet codebase: https://github.com/GiorgioMorales/hyper3dnet.) © *2020 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JRS.14.036519]

Keywords: convolution neural networks; hyperspectral classification; deep learning; parameter reduction.

Paper 200300 received Apr. 17, 2020; accepted for publication Sep. 4, 2020; published online Sep. 29, 2020.

1 Introduction

Hyperspectral images (HSIs) originated from the combination of spectroscopy and digital imaging; consequently, they consist of hundreds of narrow contiguous spectral channels, typically ranging from 400 to 2500 nm.[1] The high spectral resolution allows us to identify some materials (or their properties) by their absorption-band characteristics,[2] whereas the high spatial resolution enables us to capture details such as shapes or textures that complement the spectral information and achieve finer results.[3,4] The main reason for using HSIs is that they encompass highly complex data that can lead us to extract information that could not have been achieved using only the optical RGB bands.[5,6] Because of this, HSIs are used in a variety of applications, such as remote sensing,[7–10] agriculture,[11,12] food quality,[13] and biomedicine.[14,15]

Although HSIs have been used for decades,[16] their use has presented many challenges.[17] In general, the large number of spectral bands and the fine spectral resolution cause redundancy of the data,[18] which, in addition to the difficulty of acquiring large datasets,[19] makes it difficult to model HSI classifiers efficiently and increases the chances of overfitting.[20,21] This is a by-product of a phenomenon that is known as the "curse of dimensionality." However, it is worth noting that redundancy of information may be helpful in some applications such as information recovery.[22] Furthermore, dealing with HSIs becomes a challenge due to the increased volume of data; therefore, we should prioritize computational efficiency when storing and processing HSIs.[10] With these issues in mind, in this paper, we focus on developing a method to classify HSIs both accurately and efficiently. In a future work, we will address the problem of selecting subsets of salient spectral

[*]Address all correspondence to John W. Sheppard, E-mail: john.sheppard@montana.edu

bands for HSI classification applications exploring collinearity as a method for dimensional reduction.

In this paper, we propose a convolutional neural network (CNN) model called Hyper3DNet with two sections: a feature extractor and a spatial encoder. The feature extractor consists of a series of densely connected three-dimensional (3-D) convolutions and is used to extract deep spectral features while the spatial resolution remains the same. Afterward, the spatial encoder is composed of a series of two-dimensional (2-D) separable convolutions that reduces the spatial resolution to a one-dimensional (1-D) vector, followed by a final fully connected layer. We hypothesize that this architecture will lead us to train more efficient HSI classification models than the compared methods given a set of different HSI datasets. Here, we talk about efficiency in the sense of the performance metrics a model achieves and the number of parameters and computational operations it requires.

This paper is organized as follows. Section 2 presents some background concepts and surveys related work; Sec. 3 details the architecture of the proposed CNN; Sec. 4 provides further details about the datasets used in this work, as well as their preprocessing techniques and training details; Sec. 6 discusses the classification results shown in Sec. 5; and Sec. 7 offers concluding remarks.

2 Background

2.1 *Related Work*

Many of the classic HSI classification methods have focused on pixelwise classification in the spectral domain, not taking advantage of the spatial information. Typically, these methods are based on support vector machines (SVMs),[23,24] *k*-nearest-neighbors,[25] random forests,[26] or feed-forward neural networks.[27] Because of the high dimensionality, other methods use a feature engineering step before training a classifier.[28] This can be done using feature extraction or feature selection approaches: the former apply linear or nonlinear transformations to extract specific features from the original data,[29,30] whereas the latter select the most useful individual features (i.e., spectral bands) of the data without transforming it.[31,32] Some approaches based on SVMs using composition of kernels,[33] 3-D wavelet filters,[34] 3-D Gabor filters,[35] or conditional random fields[36] show improved classification performance, taking into account both spatial and spectral information. Their main drawback, however, is that they require hand-crafted spatial features.

Recently, deep learning techniques are being used because of their capability of learning hierarchical feature representations. CNNs are commonly used with grayscale or RGB images as inputs, which is why some remote sensing applications based on RGB satellite imagery can rely on relatively standard network architectures.[37] However, when the number of spectral bands (or channels) of the data input increases, more specialized network architectures are needed.[38] In order to exploit the spectral and the spatial information jointly, many approaches are based on 3-D CNNs, as suggested in Refs. 10 and 39. Similarly, Zhong et al.[40] designed a residual neural network (ResNet) architecture that consisted of two main sections: spectral feature learning and spatial feature learning. The first one reduces the number of original spectral bands and applies spectral kernels that affect only the spectral domain, whereas the second one uses spatial kernels that learn deep spatial representations. Due to the fact that some spectral bands can be considered as noise or simply not relevant for a certain classification task, Fang et al.[41] introduced a spectral-wise attention mechanism that emphasizes the most informative spectral features. Ma et al.[42] applied the same attention mechanism but used a double-branch 3D-CNN to extract the spectral and spatial features, respectively. Similarly, Gao et al.[43] proposed a feature boosting and suppression method in a form of an attention mechanism designed to dynamically amplify and suppress output channels from convolutional layers within a CNN. Since 2-D CNNs cannot learn channel relationship information and 3-D CNNs are more computationally complex, Roy et al.[21] proposed a hybrid-CNN model that deals with these two shortcomings. In addition to this, before training the model, they remove the spectral redundancy and reduce the dimensionality of the raw data using principal component analysis (PCA). On the other hand, other approaches use a sequence-based methodology using recurrent neural networks (RNNs),[44] a combination of CNNs and RNNs,[45] or convolutional long short-term memory.[46]

2.2 *Densely Connected Blocks*

Densely connected blocks were introduced by Huang et al.[47] to ensure maximum information flow between layers in the network. This is done by connecting all layers directly with each other using skip connections or short paths while preserving the feature-map sizes. ResNets[48] also use the idea of connecting early layers to later layers to prevent network degradation problems such as gradient vanishing or exploding when the network becomes too deep. Specifically, a densely connected block reads and transforms the state from its preceding block and passes the transformed state to the next block, along with preceding information that needs to be preserved. Unlike ResNets, a densely connected block does not add its input state to the transformed state; instead, the preservation of the preceding information of each block is ensured by the concatenation of the previous features and their transformations (Fig. 1).

2.3 *Separable Convolutions*

In 2014, Sifre[49] proposed the design of separable multidimensional convolutions, an operation that has become quite popular lately due to is ability to reduce model size and complexity.[50,51] A separable convolution consists of a depthwise convolution followed by a pointwise convolution. A depthwise convolution is a spatial operation performed independently over each input channel, whereas the pointwise convolution is a 1×1 convolution used to project the output channels into a new channel space.

Since the depthwise convolution is a channel-wise operation, if its input has a shape of $D_F \times D_F \times M$ (where D_F is the width and height of the input, and M is the number of input channels), then its kernel has a shape of $D_K \times D_K \times M$ (where D_K is the width and height of the filter, and M is the number of convolutional filters) and the output shape is $D_G \times D_G \times M$ (where D_G is the width and height of the output). The number of multiplications in one kernel operation is $D_K \times D_K \times 1 = D_K^2$, so the total number of multiplications for the kernel over the whole input channel is $D_G^2 D_K^2$. Then the computational cost of a depthwise convolution is $O(MD_G^2 D_K^2)$.

After the depthwise convolution is performed, a pointwise convolution with N filters is applied. Each kernel has the shape of $1 \times 1 \times M$, so the number of multiplications per kernel is M, and the number of multiplications per channel is $D_G^2 M$. In total, the computational cost of a pointwise convolution is $O(ND_G^2 M)$. Finally, the total computational cost of a separable convolution is $O[D_G^2 M(D_K^2 + N)]$.

In comparison, a regular convolution with an input shape of $D_F \times D_F \times M$, an output shape of $D_G \times D_G \times M$, and a kernel size of $D_K \times D_K \times N$, has a computational cost of $O(NMD_G^2 D_K^2)$. Thus compared to a regular convolution, a separable convolution reduces the computational cost by a factor of:

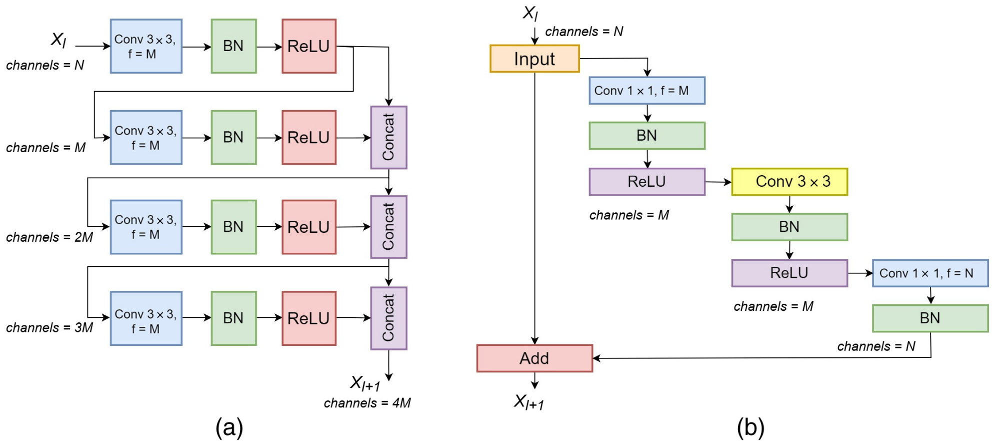


Fig. 1 Graphical examples of (a) four-layer densely connected block and (b) residual block.

$$\text{ratio} = \frac{\text{\#separable conv. mult.}}{\text{\#regular conv. mult.}} = \frac{1}{N} + \frac{1}{D_k^2}.$$

3 Proposed Framework

We propose an HSI classification method using a 3D–2D CNN architecture called Hyper3DNet. Since a HSI is basically a set of hundreds of 2-D images, we consider one HSI as a single data cube. As a consequence, the general input shape of our network is $W \times W \times D \times 1$, where W is the width and height in pixels of the input data cube and D is the spectral depth.

Figure 2 shows our proposed network architecture and its two main modules: a 3-D feature extractor and a 2-D spatial encoder. The feature extractor consists of a four-layer densely connected block, where each layer is composed of a $3 \times 3 \times 7$ 3-D convolution layer of eight filters, denoted as "CONV3D," a batch normalization layer, denoted as "BN," and a rectified linear unit activation layer, denoted as "ReLU." Notice that the difference between the diagram shown in Fig. 1 and the feature extractor shown in Fig. 2(a) is that the latter uses 3-D convolutions, which is why the concatenation ("CONCAT") is carried out along the fourth dimension so that the output of the feature extractor is a stack of 32 data cubes.

The second part of the Hyper3DNet architecture is the spatial encoder, which is shown in Fig. 2(b). Note that the input tensor coming from the 3-D feature extractor section has to be reshaped into a 3-D tensor before performing any 2-D convolution operation. Our spatial encoder takes a 3-D input and gradually compresses it into an encoded feature vector representation. This is done using four 3×3 separable convolutions of 128 filters, denoted as "SEP CONV," followed by a batch normalization layer and a rectified linear unit activation layer. The spatial dimensionality is reduced using a stride of 2 in the last three convolution operations. Finally, we reshape our output into a 1-D tensor and use a fully connected layer with softmax activation, denoted as "FC," to obtain the final classification result.

4 Experimental Design

4.1 *Datasets*

In our experiments, we used three well-known remote sensing HSI datasets: Indian Pines (IP),[52] Pavia University (PU),[53] and Salinas (SA).[54] We also experimented with the EuroSAT dataset

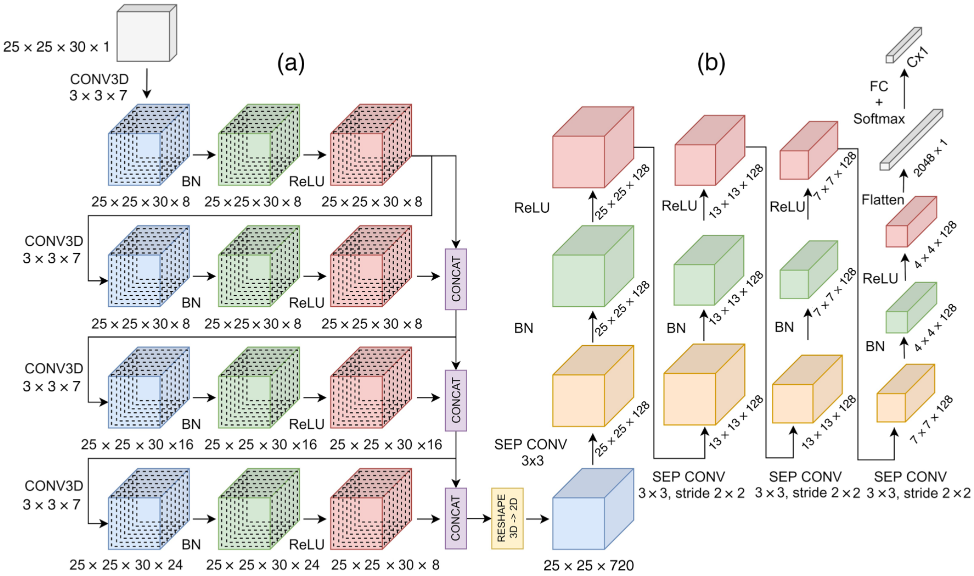


Fig. 2 Hyper3DNet architecture with a $25 \times 25 \times 30 \times 1$ HSI input. The network includes (a) a 3-D feature extractor and (b) a 2-D spatial encoder.

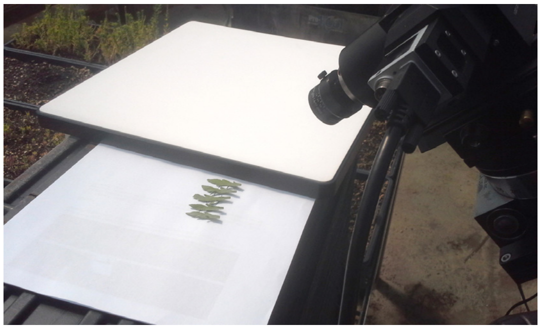


Fig. 3 A Resonon Pika L hyperspectral imager taking images of Kochia leaves in a controlled greenhouse setting.

(EU)[55] in spite of the fact that it is not a hyperspectral but a multispectral dataset, so that we validate the usefulness of our network for images with just a few spectral channels. Furthermore, we use an in-greenhouse controlled HSI dataset of Kochia leaves in order to classify three different herbicide-resistance levels (Fig. 3).

The IP dataset[52] is an aerial 145×145—pixel image of the IP site in northwestern Indiana. It was acquired using the airborne visible/infrared imaging spectrometer (AVIRIS) sensor[56] and it originally had 224 spectral bands in the wavelength range 400 to 2500 nm. The number of bands was reduced to 200 after removing 24 bands covering the region of water absorption. The data are divided into 16 classes containing agriculture, forest, and other natural perennial vegetation. Similarly, the SA dataset[54] is a 512×217—pixel aerial image of SA Valley, California, which was gathered using the same sensor as IP; thus only 200 spectral bands are used. It is divided into 16 classes containing vegetables, vineyard fields, and bare soil. On the other hand, the PU dataset is a 610×610—pixel aerial image of Pavia, northern Italy, captured with the reflective optics system imaging spectrometer sensor.[57] Unlike the other two datasets, PU is an urban dataset that has 103 spectral bands and is divided into nine classes.

The EuroSAT dataset[55] is based on Sentinel-2 satellite images covering 13 spectral bands. It consists of 27,000 64×64 images classified into 10 classes. Three of the spectral bands (i.e., aerosol, water vapor, and cirrus) do not present relevant information for our problem, so we discard them. This assumption is also supported by the fact that their spatial resolution, 60 m, is inconsistent with that of the rest of the bands: 10 and 20 m. In addition to this, Helber et al.[55] compared the contribution of each spectral band based on the classification accuracy achieved after training a ResNet50 network with each single band. Based on the results of that comparison, we decided to not use the B08A band (red edge 4), as it presents the lowest contribution. Thus for this classification task, we use only 9 out of the 13 spectral bands that this satellite acquires.

The last dataset was collected and analyzed by Scherrer et al.[27] in order to discriminate between herbicide-susceptible and herbicide-resistant biotypes of the weed Kochia (*Bassia scoparia*), which was proven to be resistant to glyphosate and dicamba, two components commonly found in commercial herbicides. A total of 76 images of Kochia with varying spatial resolution and 300 spectral bands ranging from 387.12 to 1023.5 nm were captured at the Southern Agricultural Research Center of Montana State University using the Resonon Pika L imaging system (Fig. 3).

4.2 *Data Preprocessing*

4.2.1 *Remote sensing datasets*

In preparing the datasets for analysis, first, we apply PCA on the HSI data cubes in order to reduce the number of spectral bands and remove spectral redundancy. In order to preserve spatial locality, we apply PCA at a pixel level; that is, if the original data cube has dimensions $H \times W \times Nb$ (where H is the height, W is the width, and Nb is the number of spectral bands), we reshape it to $H \times W \times Nb$ pixels. In this new reshaped tensor, each sample corresponds to a vector with Nb values, so we apply PCA to reduce its dimensions to $H \times W \times D$ pixels, where D is the desired spectral depth. Then we reshape the modified tensor to $H \times W \times D$ pixels, so that each sample returns to its original position. As a result, for the IP, PU, and SA datasets, we reduce the number of channel inputs to 30 by selecting the top 30 principal components, retaining 99.285%, 99.966%, and 99.99% of the variance, respectively. On the other hand, the EuroSAT dataset already has only nine spectral bands, so it is left without modification.

Since the IP, PU, and SA datasets consist of one big single image, we have to divide them into small patches so that each patch represents one class. Thus we extract square patches using a 25×25 pixel window around each pixel. Furthermore, we only collect patches around those pixels with an assigned label. By doing so, the new IP dataset consists of 10,249 patches; the new PU dataset, of 42,776 patches; and the new SA dataset, of 54,129 patches. The final shape of a sample from the IP, PU, and SA datasets is $25 \times 25 \times 30$ pixels; however, since the first layer of our proposed architecture requires a four-dimensional (4-D) input, we reshape each sample to $25 \times 25 \times 30 \times 1$ pixels. In other words, instead of considering the input as a stack of 30 images of 25×25 pixels, we consider it as one data cube of $25 \times 25 \times 30$ pixels. For the case of the EuroSAT, we reshape each sample to $64 \times 64 \times 9 \times 1$ pixels.

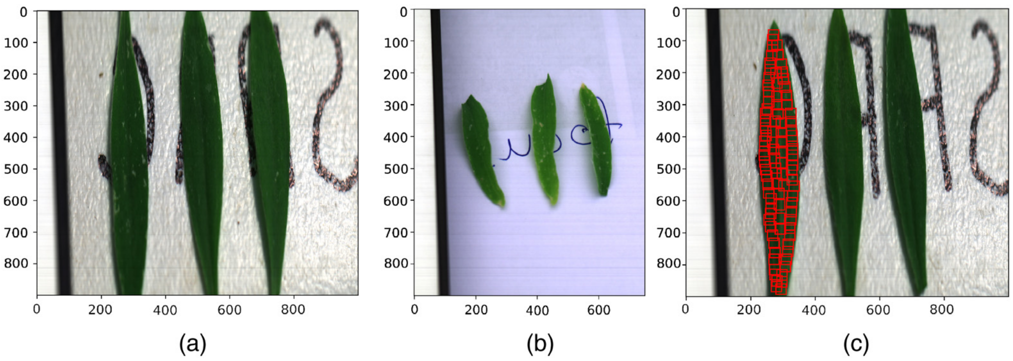
4.2.2 *Kochia dataset*

Each of the 76 collected images in the Kochia dataset has a height of 900 pixels and a width between 700 and 1200 pixels. They contain three Kochia leaves of the same herbicide-resistance class and a Spectralon panel. The white board shown in the left side of each image in Fig. 4 corresponds to the Spectralon panel, which is a nearly 100% reflective material commonly used as a Lambertian calibration reference. Therefore, the image calibration process consists of selecting the portion of image where the panel is located and calculating the mean value for each spectral band. Then we divide the image data by the calculated means; that is, we calculate the ratio of reflected light to incoming light. In this way, we calibrate the images by converting them from digital numbers to reflectance values using the Spectralon panel as a white reference.

From these images, we manually extracted 6316 25×25 pixel overlapping patches [Fig. 4(d)]. Then as with the remote sensing datasets, we apply PCA to reduce the number of spectral bands from 300 to 100 with an explained variance of 99.565%. The difference with the previous datasets is that this dataset has dimensions $6316 \times 25 \times 25 \times 300$, so we reshape it to $3,947,500 \times 300$ pixels; then we apply PCA to reduce it to $3,947,500 \times 100$ pixels; and we reshape it to $6316 \times 25 \times 25 \times 100$ pixels. Finally, similar to what we did with the remote sensing datasets, we reshape each sample to $25 \times 25 \times 100 \times 1$ pixels.

4.3 *Training Details*

All experiments are conducted on Python 3.6 on a personal computer (PC) equipped with an Intel® Xeon® CPU E5-2603 v4 at 1.70 GHz, 128 GB RAM and two NVIDIA GeForce GTX 1080 Ti graphics processing units (GPUs). The proposed CNN was trained using the Adam optimizer[58] with a learning rate of 0.0001, a momentum term β_1 of 0.9, a momentum term β_2 of 0.999, and an epsilon value of 10×10^{-8}. The minibatch sizes chosen for IP, PU, SA, EuroSAT, and Kochia datasets are 4, 4, 4, 8, and 128, respectively. Due to the fact that the last



(a)

(b)

(c)

Fig. 4 Kochia leaves at (a) 8 weeks and (b) 2 weeks. (c) Manual selection of 25×25 pixel patches (in red).

layer of our model is a softmax activation, the categorical cross-entropy function was chosen as our loss function.

We used 10-fold stratified cross validation to train and evaluate all the networks. That is, we randomly divided our datasets into 10 equally sized folds; then the training process was repeated 10 times using each random fold as an independent validation set. Furthermore, stratification indicates that each fold has the same proportion of samples of a given class. For each iteration, we train our network for 50 epochs on IP and the Kochia datasets; 40, on PU and SA; and 150, on EuroSAT.

5 Results

In order to analyze the behavior of our Hyper3DNet model statistically, we calculated four metrics on the validation sets: accuracy (OA), precision (Prec), recall (Rec), and $F1$ score. OA indicates the fraction of correctly predicted observations among total observations. Prec indicates the macroaverage precision; that is, the average precision for all the classes. The precision for class c is the fraction of correctly predicted positive observations (TP_c) among total predicted positive observations, which is defined as the sum of correctly predicted positive observations and incorrectly predicted positive observations (FP_c). Rec indicates the macroaverage recall. The recall for class c is the fraction of correctly predicted positive observations among total actual positive observations, which is defined as the sum of correctly predicted positive observations and incorrectly predicted negative observations (FN_c). The $F1$ score is the harmonic mean of Prec and Rec. The equations for all the metrics are given as follows:

$$\mathrm{OA} = \frac{\#\text{correctly classified observations}}{\#\text{total observations}},$$

$$\mathrm{Prec} = \frac{1}{\#\text{classes}} \sum_{c=1}^{\#\text{classes}} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c},$$

$$\mathrm{Rec} = \frac{1}{\#\text{classes}} \sum_{c=1}^{\#\text{classes}} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c},$$

$$F1 = 2 \cdot \frac{\mathrm{Prec} \cdot \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}.$$

The results of our proposed model are also compared to those of three supervised methods: ResNet50,[48] SpectrumNet,[59] and HybridSN.[21] ResNet50 was selected for this comparison because it constitutes a deep convolutional network that has achieved excellent generalization performance on many image classification tasks. In a similar way, HybridSN was chosen because it outperforms most of the state-of-the-art methods for HSI classification. It is worth mentioning that, for the case of the EuroSAT dataset, we applied zero-padding in the first three 3-D convolution blocks of HybridSN in order to avoid dimensionality inconsistencies. Finally, SpectrumNet was also considered for this comparison for it is a small and computationally efficient CNN, and it is also capable of being trained to process the spectral–spatial input space from multispectral satellite imagery.

Although shown to be a high-performance method, ResNet50 turns into a computationally inefficient method when it is used with such input dimensions as those of the Kochia dataset (i.e., it requires 23,912,394 parameters). Consequently, for the special case of the Kochia dataset, we decided to include the method proposed by Scherrer et al.[27] to the list of compared methods instead of ResNet50. This supervised method consists of a fully connected feedforward neural network with two hidden layers and 500 units in each layer (denoted simply as KochiaFC). Given that this network processes 1-D vectors, we modify our existing dataset as follows: first, we extract a 10×10 subpatch from the center of each patch of the Kochia dataset so that we make sure that these subpatches do not overlap; secondly, we take all the pixels of these subpatches and put them into a new 1-D vector, discarding those that have a normalized difference vegetation index <0.6.

The KochiaFC network is not used to train the other datasets because it performs pixel-wise classification; thus it will not be able to capture the underlying spatial information of a 2-D window. For instance, take a 64×64 pixel sample of class “residential building” from the EuroSAT dataset; given only the spectrum of one pixel of the window, KochiaFC will not classify it confidently into the correct class, for that pixel may be located inside a vegetation region (e.g., a tree or a garden) so that the probability of classifying it as “forest” or “pasture” could be high. Moreover, even if we select a pixel located on top of a building, the model will not have enough information to differentiate between class “residential building” and class “industrial building.”

Furthermore, in order to determine if the difference in performance scores is statistically significant, we perform a paired t-test between Hyper3DNet and the other networks with respect to the $F1$ scores. For this, we make sure that the datasets are split in the same way (i.e., using a fixed random seed) so that, for each fold, both compared methods see the same training and validation sets. In this case, the null hypothesis is that the samples of $F1$ scores were drawn from the same distribution at significance level α. For these experiments, we will consider $\alpha = 0.05$; in other words, we will reject the null hypothesis if the calculated p-value is smaller than $\alpha = 0.05$.

Table 1 shows the number of parameters and the number of floating-point operations (FLOPs) to classify a patch for each one of the compared methods on the IP, SU, SA, and EuroSAT datasets. The minimum number of parameters and MFLOPS are highlighted in bold font. Table 2 shows the same type of comparison for the special case of the Kochia dataset, where we use KochiaFC instead of ResNet50. During testing, the memory consumed by a model is used to store the outputs of intermediate layers as well as the parameters of the model. In addition, the number of parameters is related to the complexity of the model and the amount of data needed to train the model effectively. On the other hand, the number of FLOPS (multiply-and-accumulate operations) is related with the speed of processing the model.

5.1 *Results on the IP, PU, and SA Datasets*

The average classification performance and corresponding standard deviations on the IP, PU, and SA datasets are reported in Table 3. We also constructed a box plot, shown in Fig. 5, to represent

Table 1 Comparison of number of trainable parameters and MFLOPS of the different networks trained on the IP, PU, SA, and EuroSAT datasets, where the bold values indicate the minimum valued entries for that dataset.

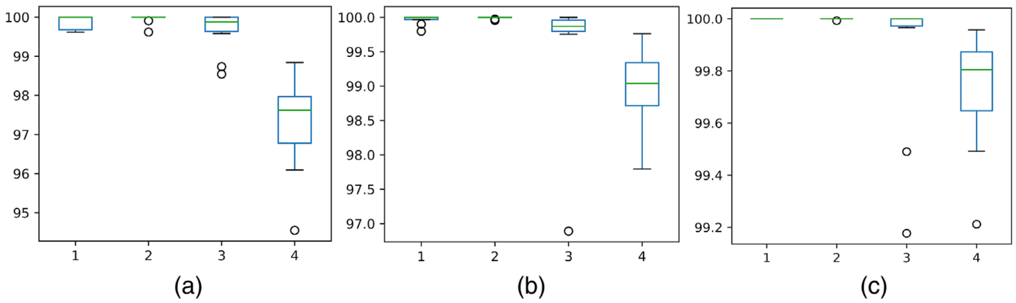
Dataset	IP and SA		PU		EuroSAT	
Method	# Param.	MFLOPS	# Param.	MFLOPS	# Param.	MFLOPS
ResNET50	23,705,168	199.56	23,637,705	199.53	23,573,898	337.15
HybridSN	5,122,176	105.44	5,121,273	105.44	15,965,562	**191.04**
SpectrumNet	741,040	**30.10**	737,449	**30.07**	729,898	205.61
Hyper3DNet	**244,328**	89.85	**228,897**	89.82	**200,322**	193.63

Table 2 Comparison of number of trainable parameters and MFLOPS of the different networks trained on the Kochia dataset, where the bold values indicate the minimum valued entries for the networks studied.

Method	# Param.	MFLOPS
KochiaFC	**402,503**	**0.8**
HybridSN	6,410,739	311.35
SpectrumNet	1,479,907	28.44
Hyper3DNet	523,483	213.73

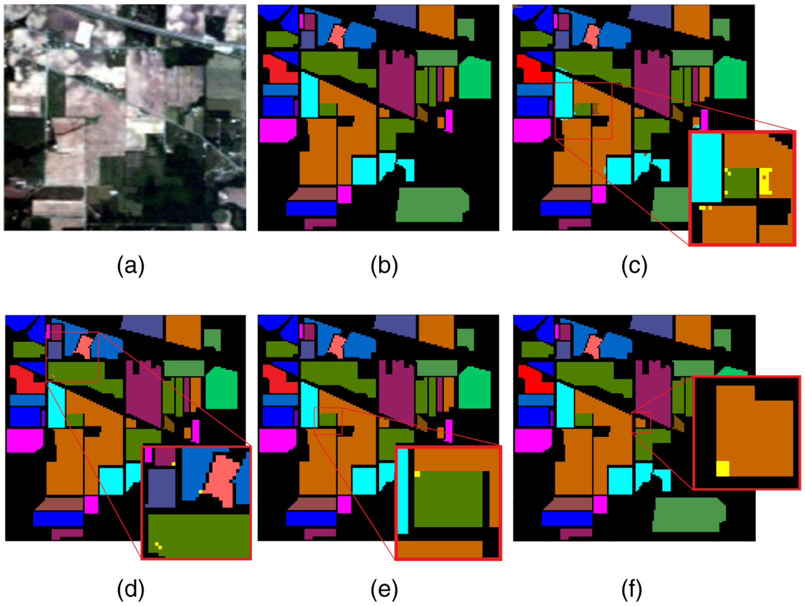
Table 3 Metrics comparison on the IP, PU, and SA datasets.

Dataset	IP				PU				SA			
Method	OA	Prec	Rec	*F*1	OA	Prec	Rec	*F*1	OA	Prec	Rec	*F*1
SpectrumNet	98.75 ± 0.39	95.95 ± 1.33	98.37 ± 0.92	96.54 ± 1.10	99.57 ± 0.23	99.21 ± 0.45	98.97 ± 0.58	99.08 ± 0.51	99.72 ± 0.23	99.48 ± 0.42	99.44 ± 0.50	99.45 ± 0.46
ResNET50	99.84 ± 0.12	99.51 ± 0.63	99.50 ± 0.65	99.47 ± 0.56	99.79 ± 0.27	99.83 ± 0.10	99.60 ± 0.51	99.71 ± 0.30	99.72 ± 0.09	99.63 ± 0.19	99.66 ± 0.11	99.64 ± 0.13
HybridSN	99.98 ± 0.04	99.98 ± 0.04	99.93 ± 0.19	99.95 ± 0.11	99.98 ± 0.04	99.98 ± 0.04	99.93 ± 0.19	99.95 ± 0.11	99.95 ± 0.04	99.93 ± 0.04	99.92 ± 0.05	99.93 ± 0.05
Hyper3DNet	99.94 ± 0.08	99.84 ± 0.26	99.89 ± 0.19	99.86 ± 0.19	99.99 ± 0.01	99.99 ± 0.02	99.98 ± 0.04	99.98 ± 0.03	99.96 ± 0.04	99.94 ± 0.06	99.94 ± 0.05	99.94 ± 0.06



(a) (b) (c)

Fig. 5 Box plot of $F1$ scores on the (a) IP, (b) PU, and (c) SA datasets. Numbers in the abscissa corresponding to (1) Hyper3DNet, (2) HybridSN, (3) ResNet50, and (4) SpectrumNet. The green line through the center of each box indicates the median value of the F1 scores. The edges of the boxes are the 25th and 75th percentiles. Whiskers extend to the maximum and minimum points. Outlier points are those past the end of the whiskers.



(a) (b) (c)

(d) (e) (f)

Fig. 6 Classification maps of the IP dataset: (a) RGB image, (b) ground-truth labels, (c) SpectrumNet, (d) ResNet50, (e) HybridSN, and (f) Hyper3DNet. Enlarged images show some misclassified pixels in yellow.

the distribution of the resulting $F1$ scores. The classification maps of different methods for the IP, PU, and SA datasets are shown in Figs. 6–8, respectively. These classification maps were generated using the validation fold with the lowest accuracy performance for each one of the datasets. Using the paired t-test, we found that the improvements of Hyper3DNet over HybridSN and ResNet50 are not statistically significant; that is, the resulting p-values are greater than $\alpha = 0.05$. However, when compared to the $F1$ scores of SpectrumNet, we achieved p-values of 8.3×10^{-5}, 5.9×10^{-4}, and 3.7×10^{-3} for IP, PU, and SA datasets, respectively, which means we have a statistical basis to reject the null hypothesis.

5.2 *Results on the EuroSAT Dataset*

Results on the EuroSAT dataset are reported in Table 4. Figure 9 shows a box plot of the distribution of the resulting $F1$ scores. We also performed paired t-tests between Hyper3DNet and HybridSN, ResNet50 and SpectrumNet, finding that the improvements of Hyper3DNet

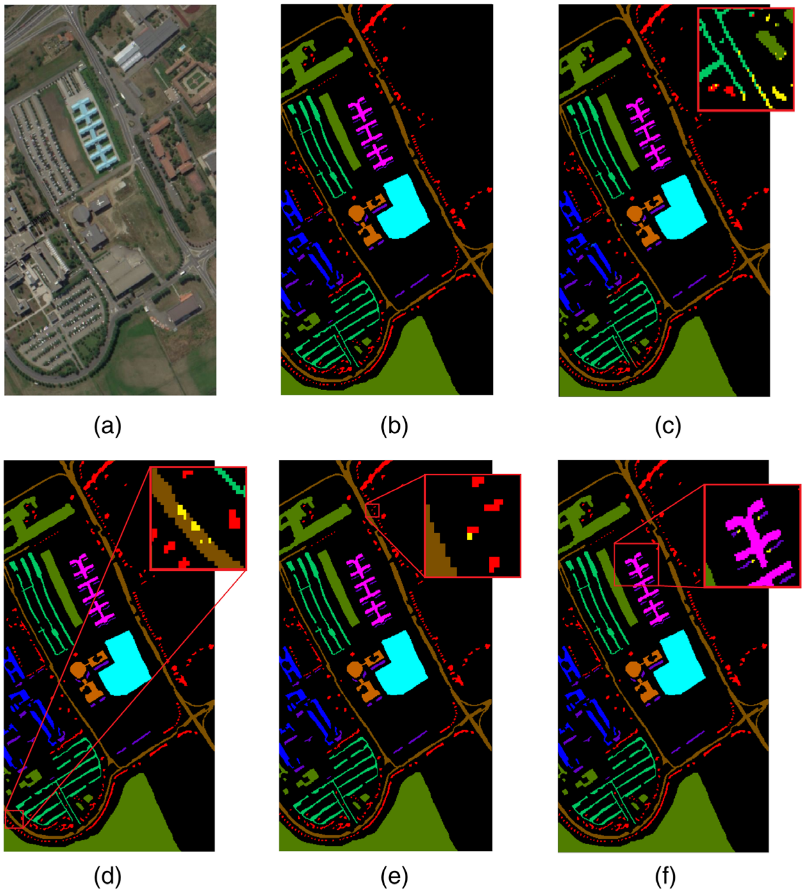


Fig. 7 Classification maps of the PU dataset: (a) RGB image, (b) ground-truth labels, (c) SpectrumNet, (d) ResNet50, (e) HybridSN, and (f) Hyper3DNet. Enlarged images show some misclassified pixels in yellow.

on $F1$ scores are statistically significant with p-values 1.9×10^{-3}, 4.2×10^{-9}, and 9.3×10^{-8}, respectively.

5.3 *Results on the Kochia Dataset*

Table 5 shows the results on Kochia dataset and Fig. 11 depicts the distribution of the $F1$ scores achieved by all the compared methods. Given that this dataset consists of only three classes, we show the four resulting confusion matrices in Fig. 10. We trained KochiaFC using our own configuration and the results are included here; however, we should note this is not a fair comparison since we did not use exactly the same dataset or input dimensions as in the other methods. Therefore, the t-test is performed between Hyper3DNet and HybridSN, and Hyper3DNet and SpectrumNet, achieving p-values of 1.60×10^{-5} and 1.40×10^{-5}, respectively, thus indicating significantly better performance by Hyper3DNet.

6 Discussion

From Tables 3–5, we conclude that our network achieves the highest performance scores regardless of which dataset is used, except for IP where HybridSN got the best results. Nevertheless, we calculated that there is no statistical significance in the difference of performance of Hyper3DNet, HybridSN, and ResNet50 when using IP, PU, and SA datasets. The reason for this is that these datasets are easy to learn, so we should focus on the other two datasets to get concluding results about our network performance. We can claim, however, that we achieve

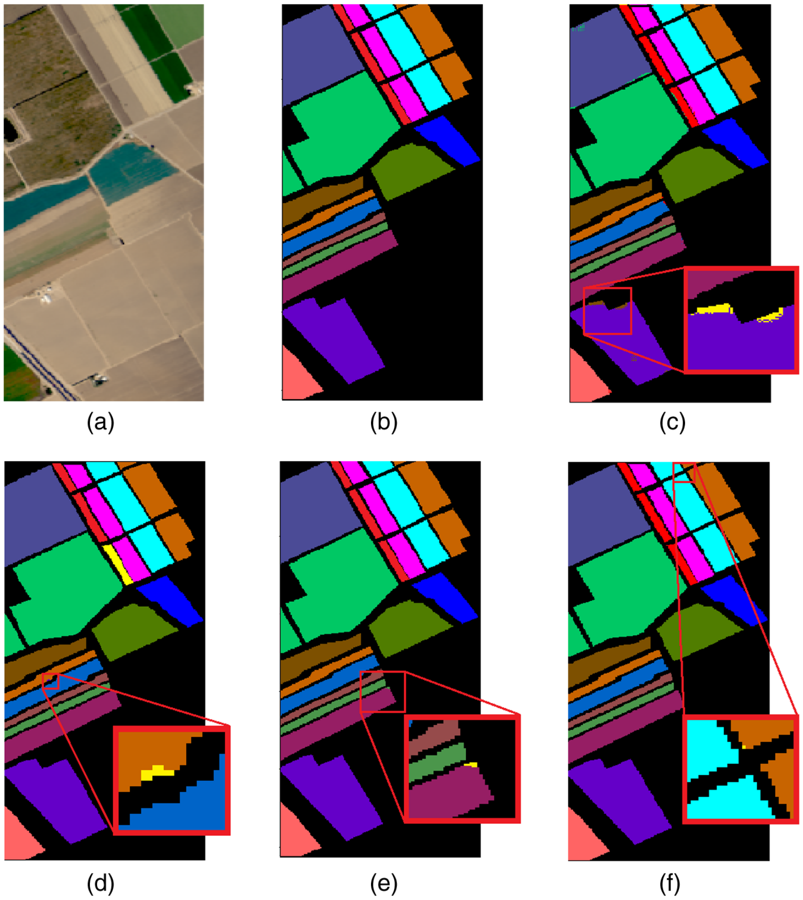


Fig. 8 Classification maps of the SA dataset: (a) RGB image, (b) ground-truth labels, (c) SpectrumNet, (d) ResNet50, (e) HybridSN, and (f) Hyper3DNet. Enlarged images show some misclassified pixels in yellow.

Table 4 Metrics comparison on the EuroSAT dataset.

Method	OA	Prec	Rec	$F1$
SpectrumNet	96.08 (±0.58)	96.08 (±0.49)	95.87 (±0.57)	95.94 (±0.56)
ResNET50	94.48 (±0.62)	94.49 (±0.52)	94.15 (±0.72)	94.24 (±0.66)
HybridSN	98.84 (±0.37)	98.85 (±0.38)	98.74 (±0.41)	98.78 (±0.39)
Hyper3DNet	99.24 (±0.15)	99.24 (±0.15)	99.19 (±0.17)	99.21 (±0.16)

comparable state-of-the-art results for these three datasets with a substantially reduced-complexity model, given that it requires fewer parameters. As a result, our model is less prone to overfitting and requires less data to train its parameters. Furthermore, Tables 4 and 5 demonstrate important performance improvements of Hyper3DNet over the other methods, which can also be verified in Figs. 9 and 11. In fact, the t-tests we performed indicate that the improvements on the observed $F1$ scores on EuroSAT and Kochia datasets are statistically significant.

From Tables 1 and 2, we note that Hyper3DNet presents the minimum number of parameters in all the cases, except for KochiaFC, without detriment to its performance, which is always the highest. It should be noted, however, that despite Hyper3DNet being a much more complex architecture than KochiaFC (two fully connected layers), they differ in the number of parameters by just 120,980. Furthermore, as stated previously, having a small number of parameters is important due to the fact that they are loaded into the memory during test time, so that reducing

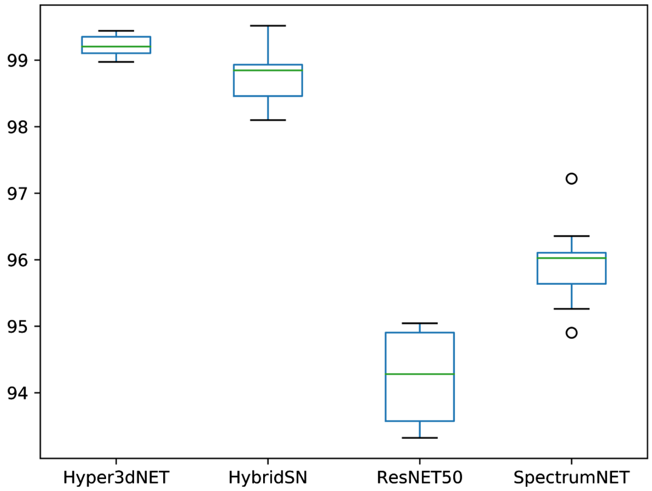


Fig. 9 Box plot of *F*1 scores of different methods on the EuroSAT dataset. The meanings of the indicators in the box are the same as Fig. 5.

Table 5 Metrics comparison on the Kochia dataset.

Method	OA	Prec	Rec	*F*1
KochiaFC	92.89 (±0.29)	93.09 (±0.30)	92.82 (±0.40)	92.95 (±0.29)
HybridSN	98.09 (±0.62)	98.18 (±0.61)	98.28 (±0.56)	98.22 (±0.58)
SpectrumNet	97.59 (±0.55)	97.81 (±0.59)	97.77 (±0.46)	97.78 (±0.51)
Hyper3DNet	99.55 (±0.22)	99.62 (±0.19)	99.57 (±0.23)	99.59 (±0.21)

this number implies fewer memory accesses and an improvement in the power efficiency.[60] For instance, the parameters of the Hyper3DNet model trained on EuroSAT require 2494 kB, whereas those of ResNet50 require 277,216 kB.

Also from Table 1, we observe that our method is the second one in terms of MFLOPS after HybridSN in the case of EuroSAT dataset, and after SpectrumNet in the case of the remaining datasets. This is explained by the fact that SpectrumNet reduces the size of the input tensor using a 2-D convolution block with a stride of 2×2 and then uses a max pooling layer after the fourth and eighth convolutional blocks. Those operations reduce the size of the outputs of the intermediate layers of the network substantially and, as a result, the number of multiply-and-accumulate operations; however, this drastic reduction causes a loss of useful spatial information that affects the performance of the network. On the other hand, when the window size of the input tensor is higher (i.e., 64×64 for EuroSAT), the spatial reduction is less drastic, thus increasing the number of MFLOPS over that of Hyper3DNet.

For both Hyper3DNet and HybridSN, the 3-D convolutions require <4 and 1 MFLOPS, respectively, because of the small number of channels or filters used (up to 24 for Hyper3DNet, see Fig. 2). The real bottleneck is located after the reshape layer used in both networks to reshape 3-D tensors into 2-D tensors. Take the example of Hyper3DNet shown in Fig. 2. After the reshaping operation, we get an output size $64 \times 64 \times 960$ (considering we are working with EuroSAT) and then we apply a 3×3 separable convolution operation of 128 filters. This operation requires 161.61 MFLOPS, which represents 83.4% of the total number of operations for the network. Similarly, the convolution operation carried out after the reshaping operation of HybridSN takes 159.48 MFLOPS, which also representes 83.4% of the total number of operations. Nevertheless, when our network requires 2.61 more MFLOPS than HybridSN for this dataset, we can argue it is a more efficient architecture because it needs 15,765,240 fewer parameters, not to mention its superior performance.

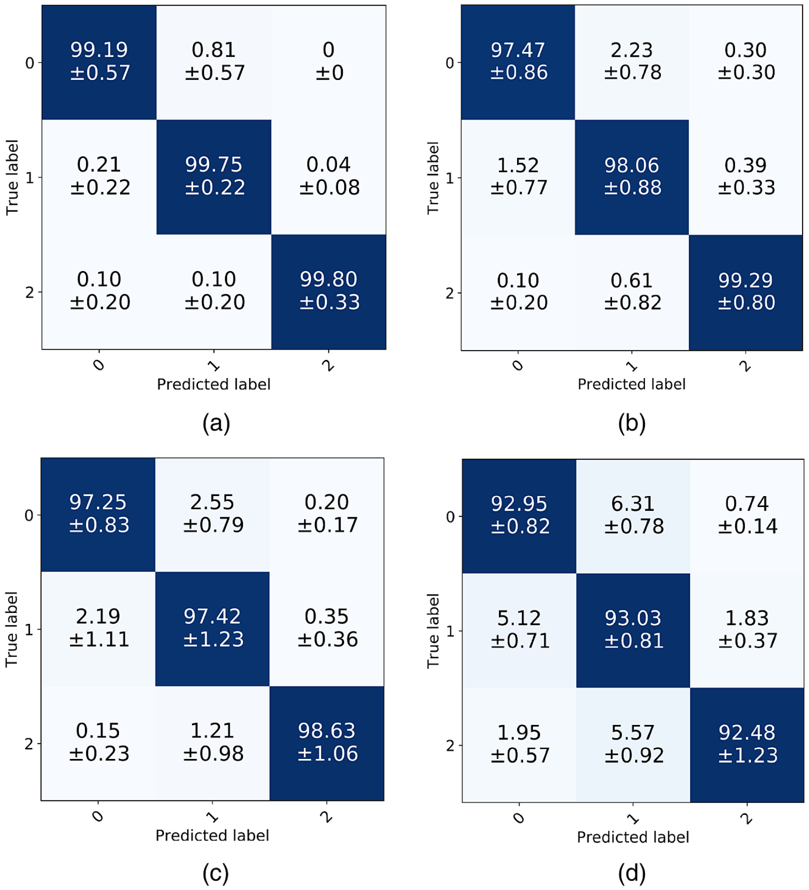


Fig. 10 Confusion matrices of different networks on the Kochia dataset: (a) Hyper3DNet, (b) HybridSN, (c) SpectrumNet, and (d) KochiaFC. Numbers in the abscissa corresponding to (0) dicamba-resistant, (1) glyphosate-resistant, and (2) herbicide susceptible.

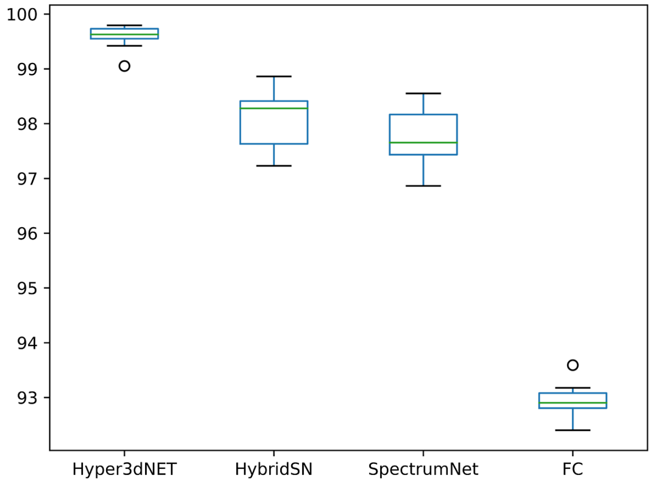


Fig. 11 Box plot of $F1$ scores of different methods on the Kochia dataset. The meanings of the indicators in the box are the same as Fig. 5.

7 Conclusions

In this paper, we have presented a deep neural network, called Hyper3DNet, for addressing the problem of HSI classification. Many previous approaches have focused solely on improving the performance metrics, while not placing enough importance on the computational cost. In contrast, our network architecture has been designed explicitly to reduce the number of trainable parameters and computational operations. The experimental results show that Hyper3DNet consistently achieves the highest classification accuracy within a variety of HSI scenarios, demonstrating that the reduced complexity of the model does not affect its performance and makes it less prone to overfitting.

Future work will focus on reducing the computational complexity of the first convolution layer of the 2-D spatial encoder immediately after the 3-D feature extractor. This is important because it represents a bottleneck, as it concentrates more than 80% of the total number of computational operations.

References

1. R. Lucas et al., *Hyperspectral Sensors and Applications*, pp. 11–49, Springer, Berlin, Heidelberg (2004).
2. R. A. Schowengerdt, “Thematic classification,” Chapter 9 in *Remote Sensing*, 3rd ed., R. A. Schowengerdt, Ed., pp. 387–456, Academic Press, Burlington (2007).
3. J. Li, J. M. Bioucas-Dias, and A. Plaza, “Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning,” *IEEE Trans. Geosci. Remote Sens.* **51**, 844–856 (2013).
4. K. Ose, T. Corpetti, and L. Demagistri, “Multispectral satellite image processing,” in *Optical Remote Sensing of Land Surface*, N. Baghdadi and M. Zribi, Eds., pp. 57–124, Elsevier, Kidlington, Oxford (2016).
5. G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *J. Biomed. Opt.* **19**(1), 010901 (2014).
6. T. Adão et al., “Hyperspectral imaging: a review on UAV-based sensors, data processing and applications for agriculture and forestry,” *Remote Sens.* **9**, 1110 (2017).
7. R. W. M. Borengasser and W. Hungate, *Hyperspectral Remote Sensing*, 1st ed., CRC Press (2007).
8. L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: a technical tutorial on the state of the art,” *IEEE Geosci. Remote Sens. Mag.* **4**, 22–40 (2016).
9. W. Zhao et al., “On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery,” *Int. J. Remote Sens.* **36**(13), 3368–3379 (2015).
10. M. Paoletti et al., “A new deep convolutional neural network for fast hyperspectral image classification,” *ISPRS J. Photogramm. Remote Sens.* **145**, 120–147 (2018).
11. A. F. A. Lowe and N. Harrison, “Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress,” *Plant Methods* **13**, 80 (2017).
12. K. Nagasubramanian et al., “Plant disease identification using explainable 3D deep learning on hyperspectral images,” *Plant Methods* **15**, 98 (2019).
13. H. Pu, L. Lin, and D.-W. Sun, “Principles of hyperspectral microscope imaging techniques and their applications in food quality and safety detection: a review,” *Compr. Rev. Food Sci. Food Saf.* **18**(4), 853–866 (2019).
14. J. Shapey et al., “Intraoperative multispectral and hyperspectral label-free imaging: a systematic review of in vivo clinical studies,” *J. Biophotonics* **12**(9), e201800455 (2019).
15. M. Halicek et al., “Optical biopsy of head and neck cancer using hyperspectral imaging and convolutional neural networks,” *J. Biomed. Opt.* **24**(3), 036007 (2019).
16. A. F. Goetz et al., “Imaging spectrometry for earth remote sensing,” *Science* **228**(4704), 1147–1153 (1985).
17. S. Li et al., “Deep learning for hyperspectral image classification: an overview,” *IEEE Trans. Geosci. Remote Sens.* **57**, 6690–6709 (2019).
18. A. Signoroni et al., “Deep learning meets hyperspectral image analysis: a multidisciplinary review,” *J. Imaging* **5**, 52 (2019).
19. M. Khodadadzadeh et al., “Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization,” *IEEE Trans. Geosci. Remote Sens.* **52**, 6298–6314 (2014).

20. D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*, Wiley, Newark, New Jersey (2005).
21. S. K. Roy et al., "Hybridsn: exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.* **17**(2), 277–281 (2020).
22. X. Li et al., "Recovering quantitative remote sensing products contaminated by thick clouds and shadows using multitemporal dictionary learning," *IEEE Trans. Geosci. Remote Sens.* **52**(11), 7086–7098 (2014).
23. F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.* **42**, 1778–1790 (2004).
24. P. W. Nugent et al., "Discrimination of herbicide-resistant Kochia with hyperspectral imaging," *J. Appl. Remote Sens.* **12**(1), 016037 (2018).
25. L. Samaniego, A. Bardossy, and K. Schulz, "Supervised classification of remotely sensed imagery using a modified k-nn technique," *IEEE Trans. Geosci. Remote Sens.* **46**, 2112–2125 (2008).
26. J. Ham et al., "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.* **43**, 492–501 (2005).
27. B. Scherrer et al., "Hyperspectral imaging and neural networks to classify herbicide-resistant weeds," *J. Appl. Remote Sens.* **13**(4), 044516 (2019).
28. X. Jia, B. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE* **101**, 676–697 (2013).
29. M. P. Uddin, M. A. Mamun, and M. A. Hossain, "Feature extraction for hyperspectral image classification," in *IEEE Region 10 Humanitarian Technol. Conf.*, pp. 379–382 (2017).
30. X. Yu et al., "Salient feature extraction for hyperspectral image classification," *Remote Sens. Lett.* **10**(6), 553–562 (2019).
31. Q. Wang, J. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Networks Learn. Syst.* **27**, 1279–1289 (2016).
32. N. S. Walton, J. W. Sheppard, and J. A. Shaw, "Using a genetic algorithm with histogram-based feature selection in hyperspectral image classification," in *Proc. Genetic and Evolutionary Comput. Conf.*, Association for Computing Machinery, Prague, Czech Republic, pp. 1364–1372 (2019).
33. M. Fauvel, J. Chanussot, and J. Benediktsson, "A spatial–spectral kernel-based approach for the classification of remote-sensing images," *Pattern Recognit.* **45**(1), 381–392 (2012).
34. X. Cao et al., "Integration of 3-dimensional discrete wavelet transform and Markov random field for hyperspectral image classification," *Neurocomputing* **226**, 90–100 (2017).
35. S. Jia, L. Shen, and Q. Li, "Gabor feature-based collaborative representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.* **53**, 1118–1129 (2015).
36. Y. Liang et al., "Hyperspectral image classification with deep metric learning and conditional random field," *IEEE Geosci. Remote Sens. Lett.* **17**(6), 1042–1046 (2020).
37. E. Maggiori et al., "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.* **55**(2), 645–657 (2017).
38. W. Hu et al., "Deep convolutional neural networks for hyperspectral image classification," *J. Sens.* **2015**, 1–12 (2015).
39. Y. Chen et al., "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.* **54**, 6232–6251 (2016).
40. Z. Zhong et al., "Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.* **56**, 847–858 (2018).
41. B. Fang et al., "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.* **11**(2), 159 (2019).
42. W. Ma et al., "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.* **11**(11), 1307 (2019).
43. X. Gao et al., "Dynamic channel pruning: feature boosting and suppression," in *Proc. 7th Int. Conf. Learn. Represent.* (2019).

44. L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.* **55**, 3639–3655 (2017).
45. H. Wu and S. Prasad, "Convolutional recurrent neural networks for hyperspectral data classification," *Remote Sens.* **9**(3), 298 (2017).
46. M. Seydgar et al., "3-D convolution-recurrent networks for spectral-spatial classification of hyperspectral images," *Remote Sens.* **11**(7), 883 (2019).
47. G. Huang et al., "Densely connected convolutional networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2261–2269 (2017).
48. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Computer Vision and Pattern Recognit.*, pp. 770–778 (2016).
49. L. Sifre, "Rigid-motion scattering for image classification," PhD Thesis, Ecole Polytechnique (2014).
50. F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1800–1807 (2017).
51. A. G. Howard et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," *CoRR*, http://arxiv.org/abs/1704.04861 (2017).
52. M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band AVIRIS hyperspectral image data set: June 12, 1992 Indian Pine Test Site 3," https://purr.purdue.edu/publications/1947/1 (2015).
53. F. Dell'Acqua et al., "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geosci. Remote Sens. Lett.* **1**(4), 322–326 (2004).
54. A. G. Gualtieri et al., "Support vector machine classifiers as applied to AVIRIS data," in *Summaries Eighth JPL Airborne Earth Sci. Workshop* (1999).
55. P. Helber et al., "Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12**, 2217–2226 (2019).
56. W. M. Porter and H. T. Enmark, "A system overview of the airborne visible/infrared imaging spectrometer (AVIRIS)," *Proc. SPIE* **0834**, 22–31 (1987).
57. B. Kunkel et al., "ROSIS imaging spectrometer and its potential for ocean parameter measurements (airborne and space-borne)," *Int. J. Remote Sens.* **12**(4), 753–761 (1991).
58. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd Int. Conf. Learn. Represent., Conf. Track Proc.*, Y. Bengio and Y. LeCun, Eds., San Diego, California (2015).
59. J. J. Senecal, J. W. Sheppard, and J. A. Shaw, "Efficient convolutional neural networks for multi-spectral image classification," in *Int. Joint Conf. Neural Networks*, Budapest, Hungary, pp. 1–8 (2019).
60. B. Wu, "Efficient deep neural networks," PhD Thesis, University of California, Berkeley (2019).

Giorgio Morales received his BSc degree in mechatronic engineering from the National University of Engineering, Lima, Peru. He is a second-year PhD student in computer science at Montana State University and a current member of the Numerical Intelligent Systems Laboratory. His research interests include image and video processing, computer vision, and machine learning algorithms with a focus on remote sensing and precision agriculture applications.

John W. Sheppard received his PhD in computer science from Johns Hopkins University and is a fellow of the Institute of Electrical and Electronics Engineers. He is a Norm Asbjornson College of Engineering distinguished professor of computer science in the Gianforte School of Computing of Montana State University. His research interests include extending and applying algorithms in deep learning, probabilistic graphical models, and evolutionary optimization to a variety of application areas, including electronic prognostics and health management, precision agriculture, and medical diagnostics.

Bryan Scherrer received his MS degrees in physics and electrical engineering from Montana State University. He is a research engineer at the Optical Remote Sensor Laboratory, Montana

State University, Bozeman, Montana. His current work focuses on the development of hyperspectral imaging and lidar systems for use in precision agriculture, ecology, and food processing.

Joseph A. Shaw received his PhD in optical sciences from the University of Arizona. He is the director of the Optical Technology Center and Norm Asbjornson College of Engineering distinguished professor of optics and electrical engineering at Montana State University, Bozeman, Montana, USA. His current work includes the development of optical remote sensing systems for use in environmental studies, ecosystems research, and precision agriculture. He is a fellow of SPIE and the Optical Society of America.