# A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification

**Article** *in* Photogrammetrie - Fernerkundung - Geoinformation · August 2020

DOI: 10.1007/s41064-020-00124-x



**2 authors:**

Bandar Alotaibi
University of Tabuk
**48** PUBLICATIONS **891** CITATIONS
SEE PROFILE

Munif Alotaibi
Shaqra University
**41** PUBLICATIONS **735** CITATIONS
SEE PROFILE

**Journal of Photogrammetry, Remote Sensing and Geoinformation Science manuscript No.**
(will be inserted by the editor)

# A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification

**Bandar Alotaibi · Munif Alotaibi**

Received: date / Accepted: date

**Abstract** Over the past few decades, hyperspectral image (HSI) classification has garnered increasing attention from the remote sensing research community. The largest challenge faced by HSI classification is the high feature dimensions represented by the different HSI bands given the limited number of labeled samples. Deep learning and convolutional neural networks (CNNs), in particular, have been shown to be highly effective in several computer vision problems such as object detection and image classification. In terms of accuracy and computational cost, one of the best CNN architectures is the Inception model i.e., the winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2014 challenge. Another architecture that has significantly improved image recognition performance is the Residual Network (ResNet) architecture i.e., the winner of the ILSVRC 2015 challenge. Inspired by the incredible performance introduced by the Inception and ResNet architectures, we investigate the possibility of combining the core ideas of these two models into a hybrid architecture to improve the HSI classification performance. We tested this combined model on four standard HSI datasets, and it shows competitive results compared with other existing HSI classification methods. Our hybrid deep ResNet-Inception architecture obtained accuracies of 95.31% on the Pavia University dataset, 99.02% on the Pavia Centre scenes dataset, 95.33% on the Salinas dataset and 90.57% on the Indian Pines dataset.

**Keywords** Hyperspectral image (HSI) classification · remote sensing · convolutional neural network (CNN) · residual neural network (ResNet) · machine learning · data mining · deep learning

**Zusammenfassung** Ein hybrides Deep ResNet- und Inception-Modell für die hyperspektrale Bildklassifikation. In den letzten Jahrzehnten ist die Aufmerksamkeit für die Klassifizierung von Hyperspektralen Bilddaten (HSI) in der Fernerkundung gestiegen. Die größte Herausforderung ist dabei die hohe Dimension an Merkmalen, die die verschiedenen HSI-Bänder angesichts der begrenzten Anzahl an Referenzdaten darstellen. Insbesondere Deep Learning und Convolutional Neural Networks (CNNs) haben sich bei verschiedenen computergestützten Visualisierungsproblemen als äußerst effektiv erwiesen. In Bezug auf Genauigkeit und Rechenaufwand ist eine der besten CNN-Architekturen das Inception-Modell, der Gewinner der ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2014. Eine weitere Architektur, die die Bilderkennung erheblich verbessert hat, ist die Residual Network (ResNet) Architektur,der Gewinner der ILSVRC 2015. Inspiriert durch die Leistung, die durch die Inception- und ResNet-Architekturen eingeführt wurde, untersuchen wir die Möglichkeit, die Kernideen dieser beiden Modelle in einer hybriden Architektur zu kombinieren, um die HSI-Klassifikation zu verbessern. Wir testeten dieses kombinierte Modell an vier Standard HSI-Datensätzen, und es zeigt sehr gute Ergebnisse im Vergleich zu anderen bestehenden HSI-Klassifikationsmethoden. Unsere hybride tiefe ResNet-Inception-Architektur erzielte Genauigkeiten von 95,31% für den Datensatz der Pavia-Universität, 99,02% für den Datensatz Pavia-Zentrum,

B. Alotaibi
Department of Computer Science and Information Technology, University of Tabuk, Tabuk, Saudi Arabia. E-mail: b-alotaibi@ut.edu.sa

M. Alotaibi
College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia. E-mail: munif@su.edu.sa

95,33% für den Salinas-Datensatz und 90,57% für die Indian Pines Daten.

## 1 Introduction

In recent years, the analysis of hyperspectral images (HSIs) obtained from distance by aircraft or satellites has received significant attention from remote sensing researchers. HSIs are represented by hundreds of band series from the electromagnetic spectrum, and they possess high spectral resolution (Mou et al, 2017). Each pixel of an HSI contains spectral information. Therefore, HSIs can be a rich source of information that allows us to find and identify objects and materials. HSIs have been used effectively in many military and civil applications. Thus, different land cover classes can be distinctly classified using supervised HSI classification. This approach has been explored for several applications such as resource management (Olmanson et al, 2013), land change monitoring (Ertürk et al, 2016), land cover mapping (Carreiras et al, 2017), water pollution detection (Garg et al, 2017) and mineral exploration (Jakob et al, 2017).

HSI classification is highly dependent on the performance of machine learning algorithms. Recently, deep learning (i.e., advanced machine learning) algorithms have been shown to be effective in HSI classification. In this paper, we introduce an effective deep learning model with the ability to improve the performance, particularly the accuracy, of existing HSI classification.

The literature on HSI classification is divided into two main categories. The first category treats every pixel's spectrum individually to assign it to a specific class. The second category is known as spectral-spatial; which combines a specific pixel with its neighboring pixels to build up a block that is input into a classifier. Our proposed work falls into the first category because we treat every pixel's spectrum individually.

Recently, deep learning has garnered the attention of HSI classification researchers (Zhang et al, 2016), (Kussul et al, 2017), (Wang et al, 2017), (Li et al, 2017b), (Zhao and Du, 2016). Deep learning capitalizes on several layers of nonlinear data processing to analyze the information patterns, extract and transform meaningful features and classify data in a supervised or nonsupervised fashion (Deng et al, 2014). Beside HSI classification, deep learning has had an enormous impact in a wide variety of fields, such as speech recognition (Abdel-Hamid et al, 2014), (Graves et al, 2013a), (Graves et al, 2013b), computer vision (Ciregan et al, 2012), (Zhong et al, 2011), natural language processing (Mesnil et al, 2015) and bioinformatics (Chicco et al, 2014), (Choi et al, 2016). Several deep learning algorithms have been proposed for HSI classification owing to the robustness of deep learning algorithms and their ability to effectively address the dense non-linearity of hyperspectral imagery.

The authors in (Chen et al, 2014, Tao et al, 2015) used an autoencoder to flatten the high-dimensional input, which consisted of local image patches, into vectors, which were input into the classifier. In contrast, Mou et al. (Mou et al, 2017) did not process HSI pixels in the usual way (as feature vectors); instead, they preprocessed the pixels through a sequential perspective to highlight the band-to-band variability and spectral correlation. They used a recurrent neural network (RNN) with gated recurrent units (GRUs) to classify the HSIs. They also presented a novel activation function called parametric rectified tanh (PRetanh) that uses high learning rates to train the neural networks equally and avoid divergence.

Deep CNNs, which are advanced types of deep learning algorithms, have been notably efficient and successful, particularly for learning visual representations. For example, Yu et al. (Yu et al, 2017) proposed a CNN model for HSI classification consisting of three convolutional layers and three normalization layers, each followed by a dropout layer and a global average pooling layer. Hu et al. (Hu et al, 2015) proposed a CNN model to classify HSIs that also consisted of three layers: a convolutional layer, a max pooling layer and a fully connected layer. Li et al. (Li et al, 2017a) combined a CNN with an extreme learning machine to perform HSI classification. Zhang et al. (Zhang et al, 2016) introduced a deep learning framework based on a CNN for HSI classification. This framework consists of three components: principal component analysis (PCA) for dimensionality reduction, a deep CNN for feature extraction and a logistic regression (LR) classifier for classification. Multiple features extracted by the PCA and CNN are combined; then, a voting method is used to classify the data. Moreover, Chen et al. (Chen et al, 2016) proposed deep learning models for HSI classification using CNNs. Several CNN architectures have been used to classify HSIs. Several convolutional and pooling layers are presented to extract the spectral-spatial deep features, which have achieved accurate classification of HSIs.

Despite the heightened performance and success of deep CNNs, several challenges remain. One is the difficulty of finding the most optimal CNN architecture. Thus, many types of architecture have been developed, such as LeNet (LeCun et al, 1998), GoogLeNet (Inception) (Szegedy et al, 2015), AlexNet (Krizhevsky et al, 2012) and VGGNet (Simonyan and Zisserman, 2014).

Most recently, Kaiming He et al. (He et al, 2016) developed a very powerful deep learning model named the deep residual network (ResNet). ResNet models are superior to other CNN models. For instance, the use of ResNet led to an improvement in the performance of three well-known HSI classification datasets (Zhong et al, 2017a), (Zhong et al, 2017b). The authors explored various ResNet architectures and determined that the best architecture consisted of two consecutive residual blocks that extract discriminative features and learn spectral and spatial information independently. The authors also investigated the effectiveness of batch normalization (BN) to address unbalanced training samples by utilizing BN for regularization. The end-to-end framework achieved state-of-the-art classification accuracy on three challenging datasets using limited training samples.

Inspired by these CNN architectures applied to image recognition (where each class is usually represented by the whole image), we combine the core principles of two of these architectures (i.e., ResNet and Inception) into a hybrid model to optimally classify HSIs (where each class is represented by a pixel with its spectral band). In contrast to the highly effective models that increase network depth to improve the performance when assigning the entire image to a class, we explore options that increasing the depth of the network reduces the performance when assigning each pixel to a specific class. The aim of this study is to propose a combination of the core ideas of ResNet and Inception architectures into one model to accurately classify HSIs. To substantiate the performance of the new approach, comparisons were made with existing approaches in terms of classification accuracy. The comparisons were conducted using four standard HSI datasets.

The continuations of this research paper are as follows:

1. To the best of our knowledge, we are the first to combine the core ideas of ResNet and Inception architectures into one model to classify hyper-spectral images.
2. Our hybrid deep ResNet-Inception model achieves promising results on four popular HSI datasets.

The rest of this paper is organized as follows. The background on CNN architectures is introduced in Section 2. The details of the proposed hybrid deep Inception and ResNet models are described in Section 3. Section 4 presents the experimental design and its results. Section 5 discusses and compares the results with those obtained using state-of-the-art methods, and Section 6 concludes the paper.

## 2 Background on ResNet and Inception Models

A typical CNN model consists of several convolutional and subsampling layers, followed by one or more fully connected (FC) layers. The FC layer is a standard multilayer neural network. The last FC layer holds the output, which is the class score. The purpose of the convolutional layer is to convolve the input image using several filters (learnable weights), and the purpose of the pooling layer is to downsample the data. Typically, the pooling layer has two types of functions: max pooling and average pooling. The CNN transforms the input image through several stacked layers from the original raw pixels to the final class scores. CNN architectures have been used as building blocks for several semantic segmentation models.

As indicated earlier, ResNet (He et al, 2016) and GoogLeNet (Szegedy et al, 2015) are powerful, advanced deep learning models. Fig. 1 (a) shows the core building block of the deep residual network. Thus, we will review ResNet and Inception architecture in this section.

### 2.1 ResNet

ResNet was proposed by Microsoft researchers in 2016, when it won the ImageNet Large Scale Visual Recognition Competition (ILSVRC), achieving an accuracy of 96.4%. The network is very deep, with 152 layers, and it features a unique architecture that introduces residual blocks, as shown in Fig. 1a to address the issue of training a deep architecture using identity skip connections. These residual blocks copy the inputs of the layers and forward them to the next layer. The identity skip connection step overcomes the issue of vanishing gradients by guaranteeing that an upcoming layer can learn something different from the input with which it is already conversant.

### 2.2 GoogleNet(Inception)

GoogleNet, which is shown in Fig. 2, is a deep CNN architecture introduced by Google researchers in 2014. This architecture won the ILSVRC as a top-5 with 93.3% accuracy. The GoogleNet's complexity is high, with 22 layers, and it introduces a novel building block known as the Inception model. This architecture does not follow the typical sequential process but uses a network in network layer, a pooling layer, and large and small convolutional layers that are computed in parallel. A 1 * 1 convolution operation is then performed for dimensionality reduction. The parallelism and the dimensionality reduction introduced in this architecture


(a) (b)

Fig. 1: (a) Core building block of deep residual networks and (b) Our-modified version of the residual block: with fully connected cascaded layers

significantly reduced the number of parameters and operations, efficiently saving memory and computational cost (Garcia-Garcia et al, 2017).

## 3 Our proposed method

In contrast to the existing deep learning methods, we propose a new CNN model that takes advantage of both the Inception model and the deep residual network ResNet model by adopting and combining the core ideas of both ResNet and Inception into one model. Deep residual networks and the Inception model have proven their ability to scale up thousands of layers while providing improved efficiency and better performance. The residual networks consist of many residual blocks with identity mapping. The Inception model is a deep convolutional network that consists of many convolutional networks.

One objective of HSI classification is to predict the land cover type by assigning and labeling individual pixels, which have many frequency bands, into individual classes. In this paper, we present a deep hyper-network architecture that can learn the deep features of HSI and provide good performance without any type of dataset augmentation or many preprocessing steps. Fig 3 shows our final proposed hybrid deep ResNet-Inception network architecture. The proposed architecture consists of two residual blocks, as shown in Fig.3.

The network has three convolutional layers followed by one average pooling layer. The outputs of each layer form the inputs to each successive layer. The architecture consists of one fully connected cascaded residual block, as shown in 1b. In this residual model, each convolutional layer receives inputs from all the previous convolutional layers. We determined the number of convolutional layers empirically and found that three convolutional layers are optimal for our model. The convolutional layers apply convolution operations to the input data, and the last pooling layer applies an average pooling operation to the data before feeding it to the classifier. We use the Adam optimization algorithm (Kingma and Ba, 2014), instead of the classical stochastic gradient descent to optimize our network because the former provides faster convergence. Furthermore, the Adam optimization algorithm is computationally efficient and less sensitive to noise. The initial learning rate is 0.001, and the batch size is set to 17 for all our datasets (the University of Pavia dataset, the Salinas dataset and the Pavia Centre scene dataset).

### 3.1 Convolutional layers

The convolutional layers apply a convolution operation to the data and then transform the data using the rectified linear unit (ReLU) function. There are three convolutional layers, each of which has 9 filters, that output 9 feature maps. The operation of each kernel (filter) is defined in equation 1

$$X^i = \varphi(W^i * X^{i-1} + \beta^i) \quad (1)$$

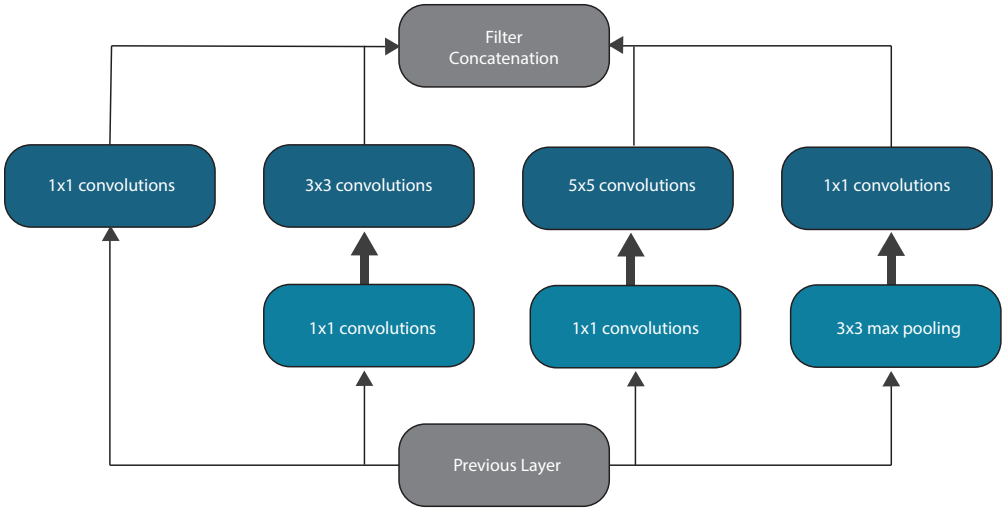



Fig. 2: Core building block of Inception model

where $*$ is the convolution operator, that convolves the filter $W^i$ with the input data $X^{i-1}$, adds the bias term $\beta$, and then applies the rectifier function $\varphi$ and outputs a feature map $X^i$.

In our proposed model, the size of each convolutional filter (1D convolution window) is 16 units, and there are 9 filters in each layer. Each filter pads the input such that the output has the same dimensions as the input tensor. The convolution stride length is one. The weights (kernels) of the convolutional layers are initialized using the Glorot uniform (Xavier) weight initialization method as suggested in (Glorot and Bengio, 2010). The bias terms are initialized to zeros.

The ReLU activation function $\varphi$ applies an elementwise operation on the input data $x$ as defined in equation 2:

$$\varphi(x) = max(x, 0) \tag{2}$$

We used 1D convolutional kernels instead of 2D or 3D kernels because the former are more suitable for the structure of the HSI data, in which each pixel and its band are represented as one vector that has one label.

Our model includes two residual models that are ultimately connected. The operation of the upper residual model is defined in equation 3:

$$\begin{aligned} X^1 &= \varphi(W^1 * X^0 + \beta^1) \\ X^2 &= \varphi(W^2 * (X^0 + X^1) + \beta^2) \\ X^3 &= \varphi(W^3 * (X^0 + X^1 + X^2) + \beta^3) \\ X^4 &= AvgP(X^3) \end{aligned} \tag{3}$$

and the lower Residual model is defined in equation 4:

$$\begin{aligned} X'^1 &= \varphi(W'^1 * X^0 + \beta^1) \\ X'^2 &= \varphi(W'^2 * (X'^0 + X'^1) + \beta^2) \\ X'^3 &= \varphi(W'^3 * (X'^0 + X'^1 + X'^2) + \beta^3) \\ X'^4 &= AvgP(X'^3) \end{aligned} \tag{4}$$

Inspired by the Inception module, we use the parallelism feature of the Inception module so that the upper and lower residual models work in parallel and are ultimately connected. The first three lines in each equation show the convolution operation for the data. We then feed the outputs of the third convolutional layers, namely, $X^3 and X'^3$ to the average pooling layer and then apply the dropout method. The methods for both average pooling and dropout are discussed in more detail in the next section.

### 3.2 Pooling layers

Our model has one pooling layer that performs average pooling with a filter size of 2 and a stride of 2. The average pooling function is defined in equation 5

$$X^i = AvgP(X^{i-1}) \tag{5}$$

where $X^{i-1}$ is the input data from the previous convolutional layer and AvgP denotes the average pooling operation.

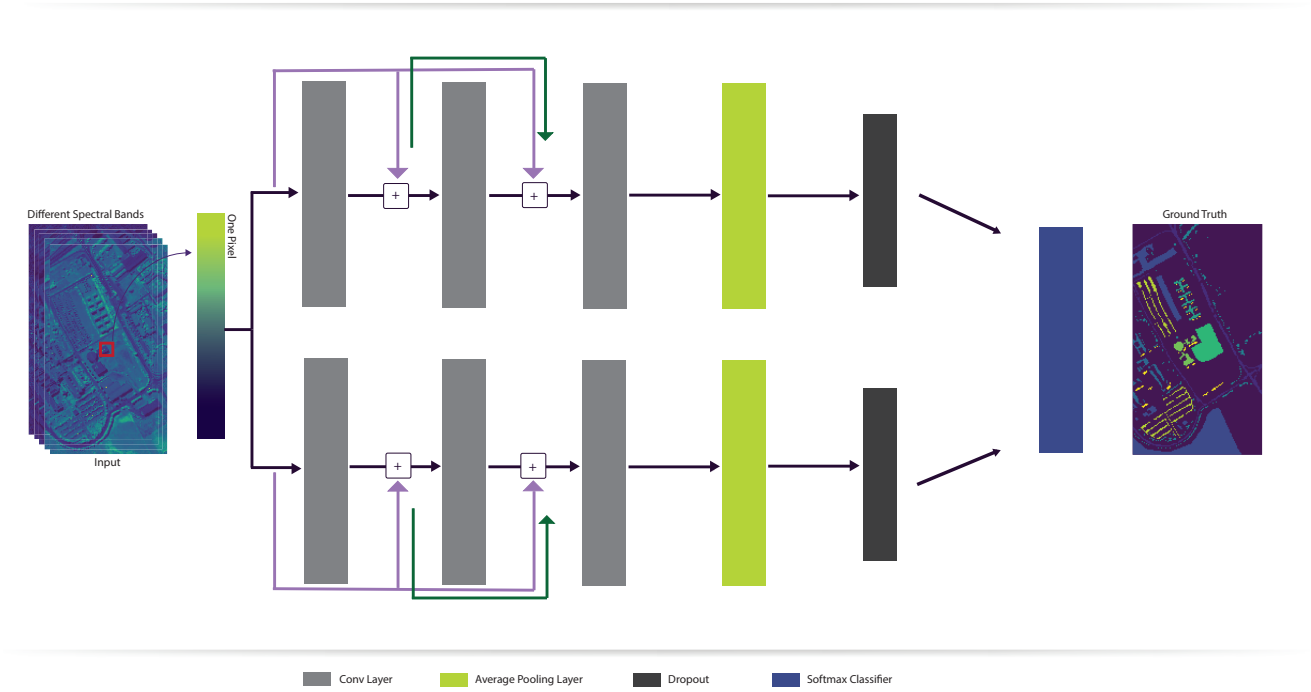
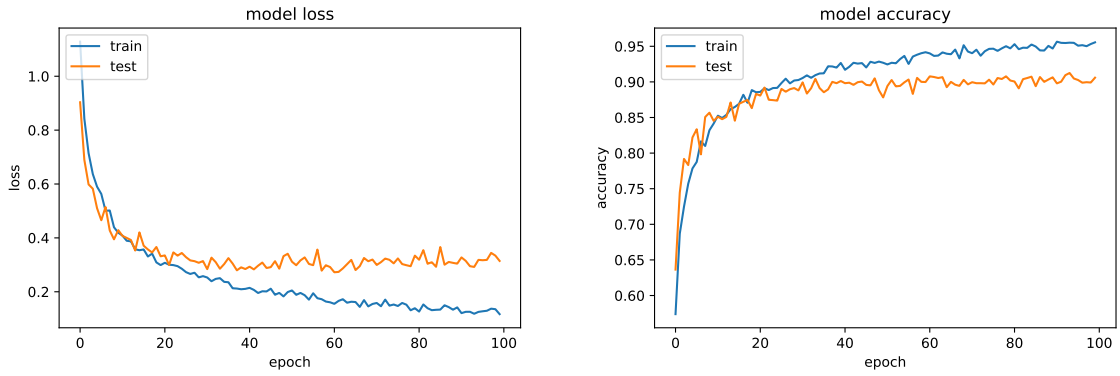



Fig. 3: Our proposed 1D hybrid deep ResNet-Inception architecture




(a) Loss (b) Accuracy

Fig. 4: (a) Loss and (b) accuracy during the convergence of our model

At the final stage, we apply a dropout technique with a probability of 0.25 directly after the average pooling layer. The output data are then input to the soft-max classifier for classification. Thus, we have no fully connected layers other than the soft-max classifier.

The first convolutional layer contains 153 trainable parameters. Each of the second and third convolutional layers contains 1,305 trainable parameters. For the University of Pavia dataset and the Pavia Centre scene dataset, the last FC layer contains 4140 trainable parameters; however, in the Salinas dataset, the last FC layer contains 14,704 parameters, and in the Indian Pines dataset, it contains 8,109 parameters. This discrepancy occurs because the University of Pavia and Pavia Centre scene datasets each have nine output classes, whereas the Salinas dataset has 16 classes, and Indian Pines has 8 output classes. Thus, the total number of trainable parameters is 8,208 in the University of Pavia and Pavia Centre scene datasets, 18,772 in the Salinas dataset and 10,872 in the Indian Pines dataset.

Fig. 4a shows the minimization of the loss function during the training phase to optimize the parameters

of our model. It also shows that our model was able to converge to a local minima within 50 epochs. Fig. 4b shows both the training and testing accuracies during our model's convergence.

## 4 Experimental Results

### 4.1 Dataset information

The Pavia University dataset was captured by a reflective optics system imaging spectrometer (ROSIS) over the city of Pavia, Italy. Each image has 610 * 340 pixels, and there are 103 bands in the dataset. The ground truth map shown in Fig. 5a consists of nine land cover classes, namely, Asphalt, Meadows, Gravel, Trees, Painted metal sheets, Bare soil, Bitumen, Self-blocking bricks and Shadows.

Table 1 shows the dataset classes and the number of samples in each class.

Table 1: Ground truth classes for the Pavia University dataset and the number of samples

| Number | Name | Samples |
|---|---|---|
| 1 | Asphalt | 6631 |
| 2 | Meadows | 18649 |
| 3 | Gravel | 2099 |
| 4 | Trees | 3064 |
| 5 | Painted metal sheets | 1345 |
| 6 | Bare soil | 5029 |
| 7 | Bitumen | 1330 |
| 8 | Self-blocking bricks | 3682 |
| 9 | Shadows | 947 |
| - | Total | 42776 |

The second dataset is the Pavia Centre scene, which was also captured by the ROSIS sensor over the city of Pavia in the northern region of Italy. The dataset consists of 102 spectral bands, and each band represents 1096 * 1096 pixels. The ground truth map shown in Fig. 5b contains nine land cover classes as well. Table 2 shows the dataset class names and the number of samples in each class.

The third dataset, named the Salinas scene, was captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the city of Salinas Valley in the state of California, USA. This dataset contains 224 bands and the ground truth map shown in Fig. 5c consists of sixteen classes including vegetables, vineyard fields, and bare soil.

Table 3 shows the class names included in the dataset and the class numbers. Note that the value zero in the ground truth maps shown in Figs 5a, 5b, 5c, and 5d represents clutter pixels, which cannot be considered a new class.

Table 2: Ground truth classes for the Pavia Centre scene dataset and the number of samples

| Number | Name | Samples |
|---|---|---|
| 1 | Water | 824 |
| 2 | Trees | 820 |
| 3 | Asphalt | 816 |
| 4 | Self-Blocking Bricks | 808 |
| 5 | Bitumen | 808 |
| 6 | Tiles | 1260 |
| 7 | Shadows | 476 |
| 8 | Meadows | 824 |
| 9 | Bare Soil | 820 |
| - | Total | 7456 |

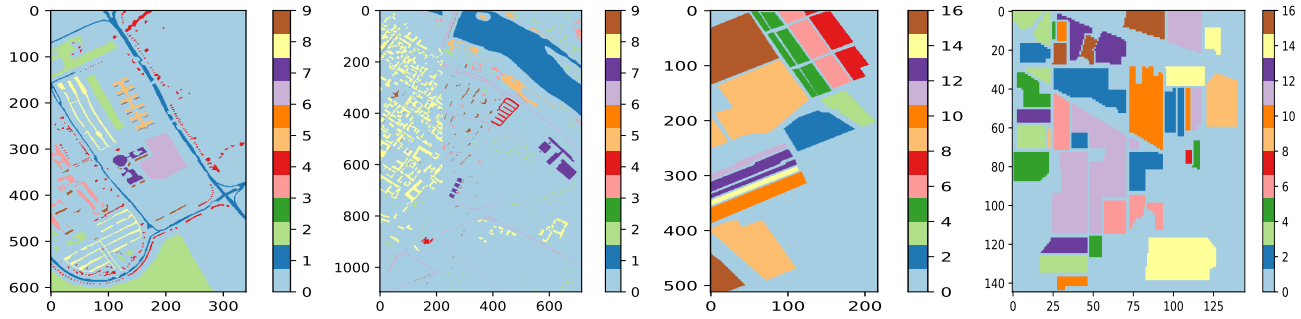
Table 3: Ground truth classes for the Salinas scene dataset and the number of samples for each class

| Number | Name | Samples |
|---|---|---|
| 1 | Broccoli green weeds 1 | 2009 |
| 2 | Broccoli green weeds 2 | 3726 |
| 3 | Fallow | 1976 |
| 4 | Fallow rough plow | 1394 |
| 5 | Fallow smooth | 2678 |
| 6 | Stubble | 3959 |
| 7 | Celery | 3579 |
| 8 | Grapes untrained | 11271 |
| 9 | Soil vineyard develop | 6203 |
| 10 | Corn senesced green weeds | 3278 |
| 11 | Lettuce romaine 4wk | 1068 |
| 12 | Lettuce romaine 5wk | 1927 |
| 13 | Lettuce romaine 6wk | 916 |
| 14 | Lettuce romaine 7wk | 1070 |
| 15 | Vineyard untrained | 7268 |
| 16 | Vineyard vertical trellis | 1807 |
| - | Total | 47929 |

The fourth dataset is Indian Pines. These data were captured by the AVIRIS sensor over northwestern Indiana. The Indian Pines dataset contains 200 bands. Table 4 shows the class names, the class numbers, and number of samples for each class in the dataset. All four datasets were downloaded from (Grupo de Inteligencia Computacional, 2014).

### 4.2 Data representation

In our proposed HSI method, each pixel and its spectral bands are treated as one sample. Thus, we treat every pixel's spectrum individually as one vector and then assign it to a specific class.



(a) Pavia University Dataset (b) Pavia Centre scene (c) Salinas Dataset (d) Indian Pines Dataset

Fig. 5: (a) Ground truth map of University of Pavia dataset, (b) Pavia centre scene, (c) Salinas dataset and (d) Indian Pines dataset

Table 4: Ground truth classes for the Indian Pines dataset and the number of samples for each class

| Number | Name | Samples |
|---|---|---|
| 1 | Corn-notill | 1428 |
| 2 | Corn-mintill | 830 |
| 3 | Grass-pasture | 483 |
| 4 | Hay-windrowed | 478 |
| 5 | Soybean-notill | 972 |
| 6 | Soybean-mintill | 2455 |
| 7 | Soybean-clean | 593 |
| 8 | Woods | 1265 |
| - | Total | 8504 |

### 4.3 Results

To evaluate the effectiveness of our proposed method in differentiating various classes of hyperspectral imagery, we used two evaluation metrics. The first metric was the overall accuracy, which can be calculated as the number of hyperspectral samples classified correctly divided by the total number of hyperspectral samples in the test set. The second metric calculated the accuracy on each class in the test set.

To demonstrate the validity of our proposed technique, we compared it with some widely used classification methods for hyperspectral imagery, namely, random forest (RF), logistic regression (LR), k-nearest neighbors (KNN), majority vote (MV), PCA for dimensionality reduction and LDA for projection and classification, the typical CNN model, and the ResNet model. RF, LR, KNN and MV were implemented using the sklearn library (Pedregosa et al, 2011), (Buitinck et al, 2013). The typical CNN and ResNet models were implemented using the keras library (Chollet et al, 2015). For the CNN-based approach, we used very similar settings and chose parameter values similar to those proposed in (Hu et al, 2015). For the ResNet model, we used four residual blocks with normalization steps as in (Zhong et al, 2017b). There is a difference between our method and the ResNet-based approach: our method uses a pixelwise approach, while the ResNet-based method uses a patchwise approach. However, to the best of our understanding and ability, we implemented ResNet as a pixelwise approach (as 1D ConvNet) to enable better comparisons with our work. We determined the best settings for other parameters such as the number of epochs of the CNN-based approach and ResNet model empirically. Next, we provide some details of the other classification techniques used in this comparison.

1. RF with 10 estimators.
2. LR using a one-vs-rest scheme.
3. KNN with the best number of neighbors in a range of 1-5.
4. MV with three classifiers in the ensemble, namely, RF, LR, and KNN.
5. PCA with 20, 40, 60, and 80 components; the best feature reduction is used and fed into LDA for supervised projection and classification.
6. Support vector machine with an RBF kernel as suggested in the literature.

The proposed method and the other state-of-the-art methods were executed using identical sets of experiments. Extensive experiments were conducted using different training and testing splits.

### 4.4 First experiment

In the first experiment, we split the samples of the datasets into 50% for training and 50% for testing. As shown in Tables 5, 6 and 7, our hybrid deep ResNet-Inception method outperformed the other approaches on the three datasets (i.e., the Pavia University dataset,

Pavia Centre scene dataset and the Indian Pines dataset) with respect to overall accuracy. To the best of our knowledge, we implemented the approaches most closely related to our technique (Hu et al, 2015), (Zhong et al, 2017b) to enable comparisons of the performances of our technique with those of similar techniques. Our method improved on the next most accurate method (i.e., the ResNet-based approach (Zhong et al, 2017b)) by more than 0.20% and on the third most accurate approach (i.e., the CNN-based approach (Hu et al, 2015)) by more than 1% on the Pavia University dataset, as shown in Table 5.

All the techniques performed well when applied to the Pavia Centre dataset, as shown in Table 6. However, our technique remained superior to the other techniques.

The ResNet-based approach (Zhong et al, 2017b) slightly outperformed our proposed method on the Salinas scene dataset. The RF-based approach was more competitive than were the other methods; however, our method was more accurate than the RF-based approach by more than 2% when tested on the Salinas scene dataset, as shown in Table 7. In addition, our technique was better than the other methods on almost all classes.

We also applied our method to the Indian Pines dataset and compared it with the most competitive methods from the other three datasets (i.e., the ResNet and the CNN-based techniques). Our technique performed very well when applied to the Indian Pines dataset, as shown in Table 8. Our method outperformed the other approaches by nearly 4.5%.

The confusion matrices (CMs) for the results of our proposed method when applied to the four datasets are shown in Figs 6a, 6b, 7a, and 7b.

### 4.5 Second set of experiments

In the next experiments, we split each of the four datasets into testing and training sets using four different percentages for training: 5%, 10%, 20%, and 30% (the remaining samples were for testing). We compared our technique with the most competitive techniques from the first experiment (i.e., the ResNet and the CNN-based approaches).The results on the four datasets are presented in Table 9. In this paper, we found that three convolutional layers were optimal, which could be due to the small size of the data. The results showed that our model still achieved good performance compared with the CNN-based approach even with a small number of training samples. As expected, the accuracies of our method and those of the other two approaches increased when we increased the training samples. When we split the datasets into 5% training and 95% testing, our proposed method performed better than the other methods on three datasets, yielding accuracies of 91.91% for Pavia University dataset, 91.64% for Salinas scene dataset and 76.79% for Indian Pines scene dataset. When we split the datasets into 10% training and 90% testing, our proposed method performed the best, achieving accuracies of 98.36% for the Pavia Centre dataset, 92.96% for the Pavia University dataset, 94.11% for the Salinas scene dataset and 79.97% for the Indian Pines scene dataset. This performance was slightly better than that of the ResNet approach on the Pavia University and the Salinas scene datasets. Moreover, our approach improved on the ResNet-based approach by approximately 14% on the Indian Pines scene dataset. However, the ResNet-based approach was slightly better than our method on the Pavia Centre dataset. Nevertheless, our method was slightly better than the CNN-based approach on the Pavia Centre dataset, was nearly 3% better on the Pavia University dataset and was considerably better than the CNN-based approach on the Salinas scene dataset (i.e., an improvement of more than 7%) and the Indian Pines dataset (i.e., an improvement of nearly 6.5%).

We split the datasets into 20% training and 80% testing sets to compare our method more strictly with both the ResNet and CNN-based approaches. The accuracies of all methods increased; however, our method's performance remained better on the four datasets, slightly better than the ResNet-based approach on the Pavia Centre scene, the Pavia University and the Salinas datasets and much better (by approximately 5.5%) on the Indian Pines scene dataset. Our method also performed slightly better than the CNN-based approach on the Pavia Centre scene and Pavia University datasets and much better on both the Salinas scene dataset (i.e., an improvement of more than 6%) and Indian Pines dataset (i.e., an improvement of nearly 6%). Finally, we split the datasets into 30% training and 70% testing to evaluate the performance of the three approaches in terms of accuracy. Our method remained more accurate than both the ResNet and CNN-based approaches on two of the datasets, while the ResNet-based approach performed slightly better than did our method on the other two datasets. Fig. 8 shows the classification results of our method, CNN-based approach, and ResNet-based approach in the four datasets when 5% of the data is used for training and 95% for testing. The three methods performed very well in term of distinguishing Pavia Centre classes. All methods decently classified the classes of both Pavia University and Salinas scene datasets, but our methods performed slightly better than the two other approaches. All methods struggled

Table 5: Accuracy of the Pavia University dataset for each class and the overall accuracy

| # | Class Name | KNN | RF | LR | MV | PCA+LDA | SVM | CNN | Deep ResNet | Our method |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Asphalt | 89.29% | 92.58% | 92.40% | 91.46% | 89.08% | 93.42 | 92.16% | 96.77% | 96.02% |
| 2 | Meadows | 97.49% | 97.91% | 97.23% | 97.94% | 93.32% | 98.89 | 96.53% | 97.35% | 98.47% |
| 3 | Gravel | 71.33% | 75.33% | 65.05% | 73.52% | 64.19% | 54.76 | 81.43% | 64.76% | 80.38% |
| 4 | Trees | 86.94% | 90.21% | 87.08% | 89.62% | 86.49% | 91.51 | 98.76% | 94.90% | 95.76% |
| 5 | Painted metal sheets | 99.25% | 99.41% | 99.85% | 100% | 99.26% | 99.85 | 99.85% | 100% | 100% |
| 6 | Bare Soil | 64.45% | 72.80% | 57.02% | 71.13% | 63.26% | 66.58 | 90.37% | 93.83% | 92.24% |
| 7 | Bitumen | 83.75% | 77.14% | 15.64% | 81.80% | 56.69% | 61.20 | 92.03% | 79.10% | 86.16% |
| 8 | Self-Blocking Bricks | 85.22% | 86.58% | 73.00% | 85.44% | 79.90% | 92.12 | 90.22% | 92.07% | 89.49% |
| 9 | Shadows | 100% | 99.79% | 99.37% | 99.79% | 99.79% | 100.0 | 100% | 99.79% | 100% |
| | **Accuracy** | 88.92% | 90.94% | 84.95% | 90.52% | 84.79% | 89.85 | 94.05% | 95.09% | **95.31**% |

Table 6: Accuracy of the Pavia Centre dataset for each class and the overall accuracy

| # | Class Name | KNN | RF | LR | MV | PCA+LDA | SVM | CNN | Deep ResNet | Our method |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Water | 100% | 99.99% | 100% | 100% | 100% | 100.0 | 100% | 100% | 100% |
| 2 | Trees | 94.58% | 96.10% | 96.50% | 96.42% | 89.71% | 94.87 | 95.87% | 96.34% | 95.81% |
| 3 | Asphalt | 87.83% | 91.65% | 87.64% | 92.30% | 91.39% | 92.36 | 97.09% | 97.28% | 95.34% |
| 4 | Self-Blocking Bricks | 83.30% | 89.49% | 61.85% | 85.54% | 76.08% | 79.60 | 87.18% | 92.47% | 90.83% |
| 5 | Bitumen | 99.50% | 95.84% | 95.20% | 97.17% | 82.38% | 97.39 | 97.44% | 97.75% | 97.48% |
| 6 | Tiles | 92.02% | 97.28% | 98.94% | 93.57% | 99.50% | 97.45 | 98.62 % | 98.07% | 98.59% |
| 7 | Shadows | 91.08% | 93.30% | 92.43% | 92.95% | 82.05% | 91.22 | 93.93 % | 96.13% | 95.64% |
| 8 | Meadows | 99.39% | 99.40% | 99.63% | 99.55% | 97.90% | 99.51 | 99.78% | 99.70% | 99.77% |
| 9 | Bare Soil | 100% | 100% | 99.79% | 100% | 100% | 100.0 | 100% | 100% | 99.86% |
| | **Accuracy** | 98.14% | 98.57% | 98.14% | 98.64% | 96.55% | 98.36 | 98.93% | 98.98% | **99.02**% |

Table 7: Accuracy of the Salinas scene dataset for each class and the overall accuracy

| # | Class Name | KNN | RF | LR | MV | PCA+LDA | SVM | CNN | Deep ResNet | Our Method |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Broccoli green weeds 1 | 98.61% | 99.30% | 99.70% | 99.20% | 99.40% | 98.90 | 99.60% | 99.70 | 99.60% |
| 2 | Broccoli green weeds 2 | 99.89% | 99.89% | 99.89% | 99.95% | 99.89% | 100.0 | 99.95% | 99.95 | 99.95% |
| 3 | Fallow | 99.49% | 99.80% | 99.09% | 99.90% | 98.28% | 99.39 | 99.59% | 100.0 | 99.90% |
| 4 | Fallow rough plow | 99.28% | 99.43% | 99.00% | 99.43% | 96.27% | 99.14 | 99.14% | 99.28 | 99.43% |
| 5 | Fallow smooth | 98.51% | 99.03% | 99.03% | 99.03% | 98.66% | 97.46 | 99.33% | 99.47 | 99.55% |
| 6 | Stubble | 100% | 99.90% | 99.95% | 100% | 99.55% | 99.90 | 99.95% | 99.95 | 99.79% |
| 7 | Celery | 99.83% | 99.78% | 99.94% | 99.83% | 99.83% | 99.72 | 99.94% | 99.94 | 99.89% |
| 8 | Grapes untrained | 77.04% | 90.92% | 89.62% | 86.32% | 85.82% | 91.59 | 97.82% | 91.41 | 92.41% |
| 9 | Soil vineyard develop | 99.71% | 99.68% | 99.90% | 99.87% | 99.97% | 99.30 | 100% | 100.0 | 100% |
| 10 | Corn senesced weeds | 94.33% | 93.96% | 93.78% | 95.67% | 93.41% | 92.31 | 97.92% | 98.23 | 97.31% |
| 11 | Lettuce romaine 4wk | 97.57% | 96.82% | 93.82% | 97.75% | 91.57% | 92.13 | 97.75% | 99.25 | 98.50% |
| 12 | Lettuce romaine 5wk | 99.58% | 99.79% | 99.48% | 99.79% | 100% | 99.79 | 100% | 99.17 | 99.79% |
| 13 | Lettuce romaine 6wk | 99.34% | 99.13% | 99.56% | 99.34% | 99.13% | 98.47 | 99.29% | 99.34 | 99.13% |
| 14 | Lettuce romaine 7wk | 97.55% | 96.07% | 96.64% | 96.82% | 91.03% | 91.59 | 99.25% | 99.06 | 99.06% |
| 15 | Vineyard untrained | 65.96% | 69.26% | 61.97% | 68.85% | 65.71% | 54.18 | 47.63% | 81.92 | 79.53% |
| 16 | Vineyard vertical trellis | 98.78% | 98.67% | 99.00% | 98.78% | 98.01% | 98.12 | 99.67% | 99.33 | 99.22% |
| | **Accuracy** | 89.91% | 93.31% | 92.03% | 92.41% | 91.32% | 90.89 | 92.21% | **95.51**% | 95.33% |

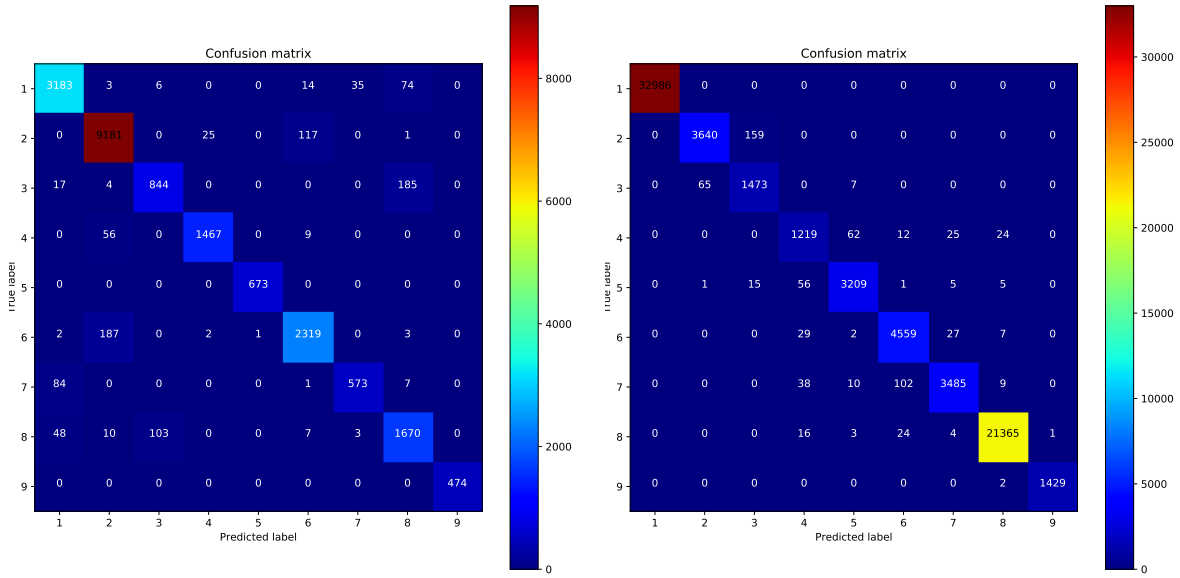
with Indian Pines dataset and the ResNet-based approach yielded the worst performance in term of classes discrimination.

The testing and training times for our model are shown in Table 10. In creating the data for this table, we used 50% of the Indian Pines dataset for training and the remaining 50% of the samples for testing. The experiments were executed on a computer with an Intel core (TM) i7-4510U CPU @ 2.00 GHz and 2.60 GHz, with 16.0 GB of RAM and a 64-bit Windows 10 operating system. The table shows the testing and training time for the typical CNN, which is faster than our method. However, during testing, both our method and the typical CNN can perform in real time.

We also performed another statistical significance test, which is McNemar's test, to prove the usefulness of our fusion method. We compare our model with the second best classifier ResNet. The two classifiers are evaluated on a single test set. We use the Indian Pines dataset and split it into 10% for training and the remaining 90% for testing. The McNemar's test employs a contingency table which is shown in Table 11. Note that under the null hypothesis, the two classifiers have the same error rate. Also, note that rejecting the null

Table 8: Accuracy of the Indian Pines dataset for each class and the overall accuracy

| # | Class Name | KNN | RF | LR | MV | PCA+LDA | SVM | CNN | Deep ResNet | Our method |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Corn-notill | 57.98% | 74.93% | 82.63% | 67.51% | 48.88% | 64.99 | 91.74% | 76.61% | 91.88% |
| 2 | Corn-mintill | 59.52% | 66.51% | 57.83% | 62.41% | 34.46% | 49.40 | 75.42% | 89.40% | 85.30% |
| 3 | Grass-pasture | 90.91% | 94.63% | 92.56% | 93.86% | 38.84% | 87.60 | 95.04% | 94.21% | 97.11% |
| 4 | Hay-windrowed | 99.16% | 98.74% | 100% | 99.58% | 100% | 98.74 | 99.16% | 99.16% | 100% |
| 5 | Soybean-notill | 69.96% | 76.95% | 73.66% | 74.69% | 46.50% | 70.16 | 88.07% | 83.74% | 84.98% |
| 6 | Soybean-mintill | 76.94% | 84.52% | 82.07% | 83.70% | 79.14% | 91.28 | 71.23% | 81.82% | 86.27% |
| 7 | Soybean-clean | 49.83% | 61.62% | 72.39% | 59.60% | 25.25% | 59.60 | 92.26% | 82.49% | 89.56% |
| 8 | Woods | 97.31% | 98.89% | 98.42% | 99.21% | 89.08% | 98.73 | 99.37% | 99.84% | 99.68% |
| | **Accuracy** | 74.44% | 82.20% | 82.20% | 79.96% | 62.59% | 79.47 | 85.58% | 86.31% | **90.57%** |




(a) CM of Pavia University

(b) CM of Pavia Centre Scene

Fig. 6: (a) Confusion Matrix of Pavia University and (b) Pavia centre scene

hypothesis means the two classifiers have a different proportion of errors.

Table 11: Contingency table

| | ResNet: Correct( | ResNet: Wrong |
|---|---|---|
| Ours: Correct | 5015 | 1311 |
| Ours: Wrong | 319 | 1009 |

The McNemar's test statistic is calculated as:

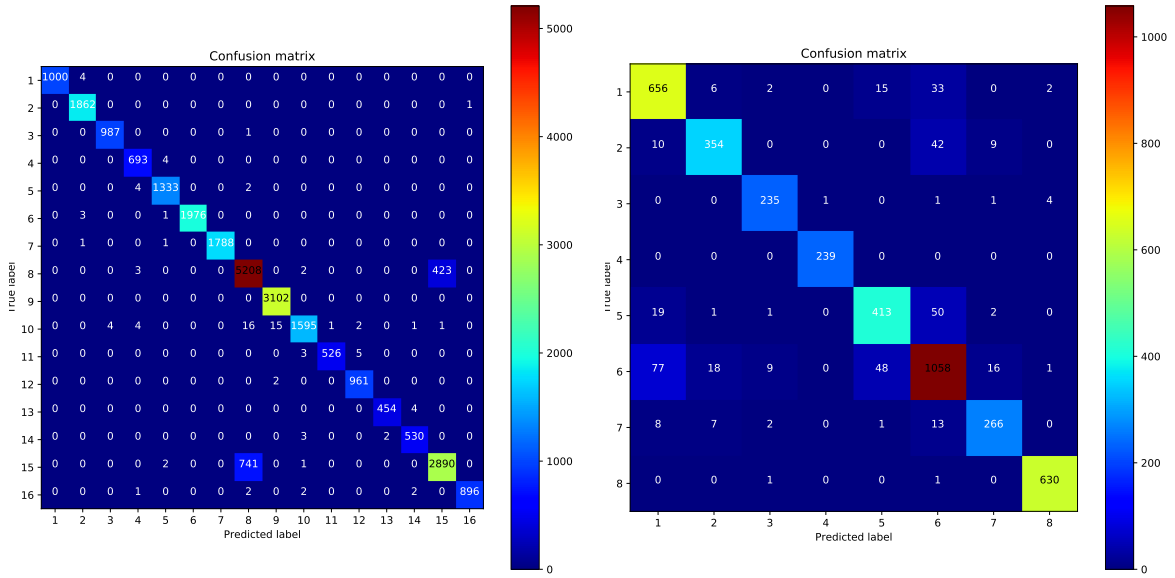
$$McNemar = (1311 - 319)^2/(1311 + 319)$$

The result of McNemar's test is 319.00 and the p-value is less than alpha at 0.005. Thus, the test rejects the null hypothesis.

## 5 Discussion

As demonstrated in the previous section, our hybrid deep ResNet and Inception architecture yielded better results than other widely used approaches for classifying HSIs. The experiments indicated that our proposed hybrid deep ResNet and Inception model is more accurate on these four datasets than both the ResNet and the typical CNN-based approaches. The results also indicated that our model outperformed other existing methods, namely, KNN, RF, LR, MV, and PCA+LDA, on these four datasets. Finally, the accuracy of our method was better than both the ResNet and CNN-based approaches on the four datasets even when the size of the training data was decreased.

Each one of the previous models has its own strength and can extract the features differently. By combining the core idea of the previous models we can take advantage of both them. We took advantage of these architec-




(a) Salinas Dataset (b) Indian Pines Dataset

Fig. 7: (a) Salinas dataset and (b) Indian Pines dataset

Table 9: Overall accuracies of the four datasets under different training split percentages

| | Training Size (Percentage) | Pavia Centre | Pavia University | Salinas scene | Indian Pines scene |
|---|---|---|---|---|---|
| Our method | 5% | 98.08% | **91.91%** | **91.64%** | **76.79%** |
| | 10% | 98.36% | **92.96%** | **94.11%** | **82.64%** |
| | 20% | **98.87%** | **93.06%** | **94.54%** | **86.11%** |
| | 30% | **98.93%** | 93.93% | 94.76% | **88.17%** |
| ResNet | 5% | **98.50%** | 91.60% | 90.91% | 62.13% |
| | 10% | **98.49%** | 92.27% | 93.25% | 69.69% |
| | 20% | 98.79% | 92.82% | 94.45% | 79.74% |
| | 30% | 98.87% | **94.28%** | **94.93%** | 84.44% |
| CNN-based approach | 5% | 97.72% | 86.59% | 90.32% | 69.05% |
| | 10% | 97.97% | 90.84% | 87.05% | 73.46% |
| | 20% | 98.24% | 91.42% | 88.27% | 80.55% |
| | 30% | 98.80% | 93.20% | 91.97% | 84.07% |

Table 10: The computation times for training and testing phases on the Indian Pines dataset

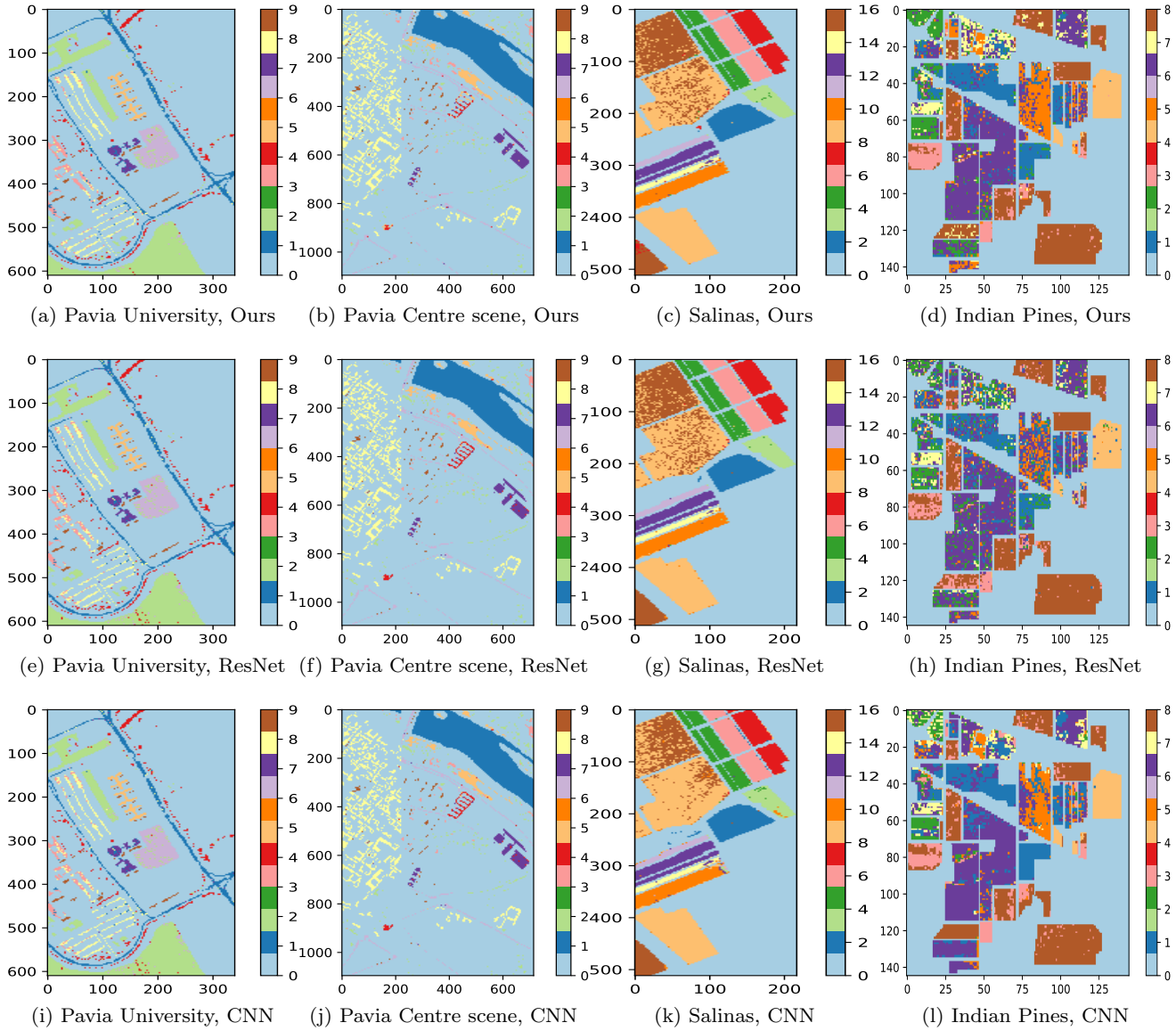
| | Training time | Testing time |
|---|---|---|
| Ours | 13.21 minutes | 3.53 seconds |
| Typical CNN | 1:28 minute | 0.39 second |

tures by combining them into one model. The core idea be- hind ResNet was its use of residual blocks (shortcut connections between layers), and the core idea behind the Inception model was its topology (layers computed in parallel and concatenated into a single output). Combining the core ideas of both models is better than each one of them individually in terms of accuracy.

The experiments also showed that our deep architecture–even with a small number of convolutional layers–can obtain impressive results on HSI classification problems. We investigated several ResNet architectures and found that using three convolutional layers achieved the best results on the four datasets. As we increased the convolutional layers, the quality of the results decreased. Based on our extensive experiments, increasing the depth of the network does not improve the performance due to the small number of labeled training samples. Thus, the proposed hybrid deep ResNet-Inception model, with its identity mappings, approximates complex non-linear functions from high-dimensional input data and shows good performance.

## 6 Conclusion

HSI classification is a popular research topic among remote-sensing researchers owing to its ability to iden-



(a) Pavia University, Ours (b) Pavia Centre scene, Ours (c) Salinas, Ours (d) Indian Pines, Ours

(e) Pavia University, ResNet (f) Pavia Centre scene, ResNet (g) Salinas, ResNet (h) Indian Pines, ResNet

(i) Pavia University, CNN (j) Pavia Centre scene, CNN (k) Salinas, CNN (l) Indian Pines, CNN

Fig. 8: (a, e, and i) are the classification maps of Pavia University dataset for our approach, ResNet-based method, and the typical CNN, respectively. (b, f, and j) are the classification maps of Pavia Centre scene dataset for our approach, ResNet-based method, and the typical CNN, respectively. (c, g, and k) are the classification maps of Salinas dataset for our approach, ResNet-based method, and the typical CNN, respectively. (d, h, and l) are the classification maps of Indian Pines as as produced by our approach, ResNet-based method, and the typical CNN, respectively.

tify objects from a long distance. Many machine learning algorithms have been evaluated previously for their capacity to efficiently classify HSI. Some of these algorithms have demonstrated limitations with respect to accuracy, and others have been associated with high computational costs. Recently, deep CNN architecture has been shown to achieve superior performance regarding HSI classification. Inception and ResNet are two examples of powerful CNN architecture for image recognition, where an entire image is assigned to a specific class. We combined the core concepts of these two types of architecture into a single hybrid model to classify HSIs, where each pixel represented a class. The hybrid deep ResNet–Inception architecture was tested using four HSI datasets, and it provided better results than other widely used approaches for classifying HSIs.

**Acknowledgements** This research was financially supported by the Deanship of Scientific Research, University of Tabuk, Tabuk, Saudi Arabia under grant number S-0181-1439.

## References

Abdel-Hamid O, Mohamed Ar, Jiang H, Deng L, Penn G, Yu D (2014) Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 22(10):1533–1545

Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G (2013) API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp 108–122

Carreiras JM, Jones J, Lucas RM, Shimabukuro YE (2017) Mapping major land cover types and retrieving the age of secondary forests in the brazilian amazon by combining single-date optical and radar remote sensing data. Remote Sensing of Environment 194:16–32

Chen Y, Lin Z, Zhao X, Wang G, Gu Y (2014) Deep learning-based classification of hyperspectral data. IEEE Journal of Selected topics in applied earth observations and remote sensing 7(6):2094–2107

Chen Y, Jiang H, Li C, Jia X, Ghamisi P (2016) Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing 54(10):6232–6251

Chicco D, Sadowski P, Baldi P (2014) Deep autoencoder neural networks for gene ontology annotation predictions. In: Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM, pp 533–540

Choi E, Schuetz A, Stewart WF, Sun J (2016) Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association 24(2):361–370

Chollet F, et al (2015) Keras

Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, pp 3642–3649

Deng L, Yu D, et al (2014) Deep learning: methods and applications. Foundations and Trends® in Signal Processing 7(3–4):197–387

Ertürk A, Iordache MD, Plaza A (2016) Sparse unmixing-based change detection for multitemporal hyperspectral images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9(2):708–719

Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J (2017) A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:170406857

Garg V, Kumar AS, Aggarwal S, Kumar V, Dhote P, Thakur PK, Nikam BR, Sambare R, Siddiqui A, Muduli PR, et al (2017) Spectral similarity approach for mapping turbidity of an inland waterbody. Journal of Hydrology

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp 249–256

Graves A, Jaitly N, Mohamed Ar (2013a) Hybrid speech recognition with deep bidirectional lstm. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, IEEE, pp 273–278

Graves A, Mohamed Ar, Hinton G (2013b) Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, IEEE, pp 6645–6649

Grupo de Inteligencia Computacional (2014) Ehu. URL http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Hu W, Huang Y, Wei L, Zhang F, Li H (2015) Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors 2015

Jakob S, Zimmermann R, Gloaguen R (2017) The need for accurate geometric and radiometric corrections of drone-borne hyperspectral data for mineral exploration: Mephysto—a toolbox for pre-processing drone-borne hyperspectral data. Remote Sensing 9(1):88

Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Kussul N, Lavreniuk M, Skakun S, Shelestov A (2017) Deep learning classification of land cover and crop types using remote sensing data. IEEE Geoscience and Remote Sensing Letters 14(5):778–782

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recog-

nition. Proceedings of the IEEE 86(11):2278–2324
Li Y, Hu J, Zhao X, Xie W, Li J (2017a) Hyperspectral image super-resolution using deep convolutional neural network. Neurocomputing
Li Y, Zhang H, Shen Q (2017b) Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. Remote Sensing 9(1):67
Mesnil G, Dauphin Y, Yao K, Bengio Y, Deng L, Hakkani-Tur D, He X, Heck L, Tur G, Yu D, et al (2015) Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23(3):530–539
Mou L, Ghamisi P, Zhu XX (2017) Deep recurrent neural networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing
Olmanson LG, Brezonik PL, Bauer ME (2013) Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: The mississippi river and its tributaries in minnesota. Remote Sensing of Environment 130:254–265
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
Tao C, Pan H, Li Y, Zou Z (2015) Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. IEEE Geoscience and remote sensing letters 12(12):2438–2442
Wang L, Zhang J, Liu P, Choo KKR, Huang F (2017) Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. Soft Computing 21(1):213–221
Yu S, Jia S, Xu C (2017) Convolutional neural networks for hyperspectral image classification. Neurocomputing 219:88–98
Zhang L, Zhang L, Du B (2016) Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geoscience and Remote Sensing Magazine 4(2):22–40
Zhao W, Du S (2016) Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. IEEE Transactions on Geoscience and Remote Sensing 54(8):4544–4554
Zhong Sh, Liu Y, Liu Y (2011) Bilinear deep learning for image classification. In: Proceedings of the 19th ACM international conference on Multimedia, ACM, pp 343–352
Zhong Z, Li J, Luo Z, Chapman M (2017a) Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework. IEEE Transactions on Geoscience and Remote Sensing
Zhong Z, Li J, Ma L, Jiang H, Zhao H (2017b) Deep residual networks for hyperspectral image classification. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Institute of Electrical and Electronics Engineers