

A Novel Hyperspectral Image Classification Technique Using Deep Multi-Dimensional Recurrent Neural Network

K. Thilagavathi^{1,*} and A. Vasuki²

¹ Department of Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore-641 049, Tamilnadu, India.

² Department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore-641 049, Tamilnadu, India.

Received: 2 Apr. 2019, Revised: 11 May 2019, Accepted: 18 May 2019

Published online: 1 Nov. 2019

Abstract: In this paper, a novel Multi-Dimensional Recurrent Deep Neural Network is proposed for classifying hyperspectral images. Deep Learning Networks have developed rapidly with applications in several fields including computer vision, healthcare, bioinformatics and machine learning. Multi-Dimensional Recurrent Deep Neural Networks are a special case of directed acyclic graph networks in which standard Recurrent Neural Networks are realized by giving recurrent connections along all spatio-temporal dimensions of the data and the recurrent connection size is equal to the dimension of the data. In this work, two Recurrent Neural Networks are replaced by one Multi-Dimensional Recurrent Deep Neural Network to learn middle-level visual patterns and spatial dependencies between them. In the last stage, fully connected layers are used to learn a global image representation. Due to the recurrent connections, this method is robust to local distortions such as image rotation and shear. Without suffering from scaling problems, it brings additional advantages over Recurrent Neural Networks to multi-dimensional data. This paper investigates hyperspectral image classification with the proposed network and the results have been validated with hyperspectral datasets namely Pavia University and Salinas images. There is an improvement in the classification accuracy of this newly proposed method in comparison to classical methods like Support Vector Machine, Convolutional Neural Network (CNN) and Recurrent Convolutional Neural Network (RCNN).

Keywords: Hyperspectral, Image Classification, Convolutional Neural Network, Recurrent Neural Network, Multi-Dimensional Recurrent Neural Network.

1 Introduction

Hyperspectral Image (HSI) encompasses spectral as well as spatial information. In contrast to multispectral images, hyperspectral images have hundreds of spectral bands that make it feasible to distinguish objects that closely resemble each other. HSI plays a major role in applications like precision agriculture [1], land-use monitoring [2], mining [3], space exploration, defense [4], change detection [5], environment measurements, and so forth.

Recently, Deep Learning (DL) with Convolutional Neural Networks (CNN) has shown promising results for HSI classification [6]. The ability of extracting the spatial contextual information makes the CNN classifier adaptive. But it fails for pixel-based processing of HSI, due to its 2D filter processing characteristic. To overcome this, Recurrent Neural Network (RNN) can be used as

another deep learning tool for HSI classification [7]. RNNs use Long Short Term Memory (LSTM) and has found to be time consuming in training and managing long-sequences. The combination of CNN with RNN had been investigated as RCNN model for HSI classification [8]. By using spatial contextual feature, CNN and RNN layers extract the local invariant information among spectral bands patch-level wise. Multi-Dimensional Recurrent Neural Network (MDRNN) is another deep learning network investigated in the literature for spectral applications, but not yet used for hyperspectral remote sensing images.

Various methods, either spectral or spatial information or both, have been applied for processing and classifying remotely-sensed images efficiently [9]. This includes unsupervised methods like clustering [10] and supervised methods like Support Vector Machine (SVM). Supervised classifiers are preferred for their high classification

* Corresponding author e-mail: thilaga.kct@yahoo.com

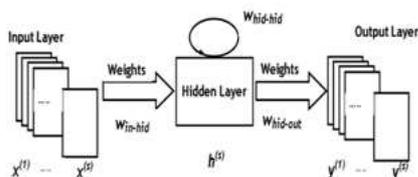


Fig. 1: General RNN structure showing input layer, hidden layer and output layer

accuracies, despite the limitations from the lack of a large number of training samples. As HSIs possess high-dimensional information, it often leads to Hughes phenomenon (curse of dimensionality) [11]. Hence, the supervised schemes have some specific challenges in the field of classification. As SVMs [12] and multi-nomial Logistic Regression (MLR) [13] are capable of managing large input spaces, they give better performance for the classification of HSI. Semi-supervised and active learning schemes [14] are recommended for classification in which the number of training samples is restricted.

Artificial Neural Networks (ANNs) are preferred to classify linearly non-separable data [15]. CNNs have been found to be effective for image classification, as they combine the spectral and spatial characteristics of data efficiently.

Stacked Auto Encoder (SAE) [16] and Deep Belief Network (DBN) [17] are the earlier networks in this category. Recent research with CNN and RNN-based models is available in literature [18–21].

2 Recurrent Neural Network

In RNNs, contextual information of the image sequences is encoded by connecting current states with the output of previous hidden states in the form of feedback loops.

A typical RNN [22] is illustrated in Fig. 1, where, the input layer of length S is $x^{(s)}$, the hidden layer is $h^{(s)}$ and the predicted output is $y^{(s)}$. Here S is the set of states, i.e., $S \in [1, \dots, S]$. The definition of hidden layer and RNN output is as follows:

$$h^{(s)} = f_h \left(W_{hh}h^{(s-1)} + W_{ih}x^{(s)} + b_h \right) \quad (1)$$

$$y^{(s)} = f_o \left(W_{ho}h^{(s)} + b_o \right) \quad (2)$$

where

$x^{(s)}$ is the input,

$h^{(s)}$ is the hidden layer,

$y^{(s)}$ is the output,

W_{ih} , W_{hh} and W_{ho} are the transformation matrices between input to hidden, hidden to hidden layers and hidden to output

b_h and b_o are the constant bias terms and

f_h and f_o are the non-linear activation functions.

By updating the weights W_{ih} , W_{hh} and W_{ho} , RNNs will retain information about the data processed earlier. RNN will learn spatial dependencies at distinct spatial spots and the same will be available in the connections between image regions. In general, one dimensional data are appropriate for RNN and hence the output from CNN is converted to four one-dimensional data to process them with the RNN. These four one-dimensional data will provide the context in all image regions.

Referring to the above equation, the four one-dimensional RNNs are defined as:

$$\begin{aligned} h_{\rightarrow}^{(s)} &= f_h \left(W_{hh\rightarrow}h_{\rightarrow}^{(s-1)} + W_{ih\rightarrow}x^{(s)} + b_{h\rightarrow} \right) \\ h_{\leftarrow}^{(s)} &= f_h \left(W_{hh\leftarrow}h_{\leftarrow}^{(s-1)} + W_{ih\leftarrow}x^{(s)} + b_{h\leftarrow} \right) \\ h_{\downarrow}^{(s)} &= f_h \left(W_{hh\downarrow}h_{\downarrow}^{(s-1)} + W_{ih\downarrow}x^{(s)} + b_{h\downarrow} \right) \\ h_{\uparrow}^{(s)} &= f_h \left(W_{hh\uparrow}h_{\uparrow}^{(s-1)} + W_{ih\uparrow}x^{(s)} + b_{h\uparrow} \right) \end{aligned} \quad (3)$$

where:

$h_{\rightarrow}^{(s)}$ is the left side-to-right side hidden layer units,

$h_{\leftarrow}^{(s)}$ is the right side-to-left side hidden layer units,

$h_{\downarrow}^{(s)}$ is the top side-to-bottom side hidden layer units and

$h_{\uparrow}^{(s)}$ is the bottom side-to-top side hidden layer units. The hidden layer is a summation of the above four equations:

$$h^{(s)} = h_{\rightarrow}^{(s)} + h_{\leftarrow}^{(s)} + h_{\downarrow}^{(s)} + h_{\uparrow}^{(s)} \quad (4)$$

The weights of the one-dimensional RNN sequences are updated in forward and backward procedures and are as follows:

$$\begin{aligned} W_{ih\rightarrow}^{(\delta+1)} &= W_{ih\rightarrow}^{(\delta)} + x^{(\delta)} e_{h_{\rightarrow}}^{(\delta)} \epsilon \\ W_{hh\rightarrow}^{(\delta+1)} &= W_{hh\rightarrow}^{(\delta)} + h_{\rightarrow}^{(\delta-1)} e_{h_{\rightarrow}}^{(\delta)} \epsilon \\ W_{ih\leftarrow}^{(\delta+1)} &= W_{ih\leftarrow}^{(\delta)} + x^{(\delta)} e_{h_{\leftarrow}}^{(\delta)} \epsilon \\ W_{hh\leftarrow}^{(\delta+1)} &= W_{hh\leftarrow}^{(\delta)} + h_{\leftarrow}^{(\delta-1)} e_{h_{\leftarrow}}^{(\delta)} \epsilon \\ W_{ih\downarrow}^{(\delta+1)} &= W_{ih\downarrow}^{(\delta)} + x^{(\delta)} e_{h_{\downarrow}}^{(\delta)} \epsilon \\ W_{hh\downarrow}^{(\delta+1)} &= W_{hh\downarrow}^{(\delta)} + h_{\downarrow}^{(\delta-1)} e_{h_{\downarrow}}^{(\delta)} \epsilon \\ W_{ih\uparrow}^{(\delta+1)} &= W_{ih\uparrow}^{(\delta)} + x^{(\delta)} e_{h_{\uparrow}}^{(\delta)} \epsilon \\ W_{hh\uparrow}^{(\delta+1)} &= W_{hh\uparrow}^{(\delta)} + h_{\uparrow}^{(\delta-1)} e_{h_{\uparrow}}^{(\delta)} \epsilon \end{aligned} \quad (5)$$

where $e_h^{(\delta)}$ is the gradient of error,

δ is the step size and ϵ is the learning rate.

Two fully connected layers, defined below, are used to collect all hidden units in the RNN layers

$$g = f_g (W_{hg}H + b_g) \quad (6)$$

$$y = f_y (W_{gy}g + b_y) \quad (7)$$

$$H = \left[\left(h^{(1)} \right)^T, \dots, \left(h^{(s)} \right)^T, \dots, \left(h^{(S)} \right)^T \right]^T \quad (8)$$

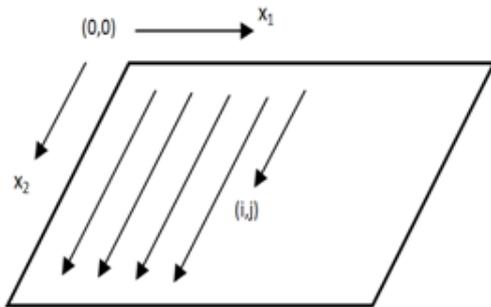


Fig. 2: 2D sequence ordering of MDRNN with forward pass begins at (0, 0) and follows the direction of arrows

where W_{hg} is the transfer matrix between the concatenated RNN outputs H and the global hidden layer g , W_{gy} is the transfer matrix between g and the predicted class label y , H is the sequential links of all sequential states $h^{(s)} (s = 1, \dots, S)$, b_g & b_y are the bias values and f_g & f_y are non-linear activation function and softmax function.

3 Multi-dimensional Recurrent Neural Network

MDRNNs [23] are formed by furnishing recurrent connections along all spatio-temporal dimensions of the data in standard RNNs. Due to these connections, MDRNN is capable of managing local variation due to rotations and shears. When dealing with multi-dimensional data, MDRNN is more advantageous than RNNs and it does not affect scaling.

The single recurrent connection is substituted by n number of recurrent connections in MDRNN if the dimension of the data is n . At each point in the input sequence, if the input and all the previous hidden layer data are fed forward, it is called forward pass (Fig. 2). On the other hand, at the hidden layer, if the output error derivatives and its ‘future’ derivatives are fed back, it is called backward pass.

Error gradient of multi-dimensional network is calculated by taking the derivative of cost function with respect to weights. x_j^m is the input to the j^{th} unit and h_k^m is the activation of the k^{th} hidden unit at a point $m = (m_1, \dots, m_n)$ in an n -dimensional sequence x . The recurrent connection from unit j to unit k will have w_{jk}^d as weight along the dimension d . The algorithm for forward pass is shown in Algorithm 1 [24].

In Algorithm 1, there are I input units, K output units and H summation units for an n -dimensional MDRNN. The dimensions are $D_1 D_2, \dots, D_n$, and ϕ_k is the activation function of the hidden unit k . The algorithm for the backward pass is shown in Algorithm 2 [24].

Algorithm 1: MDRNN forward pass

```

for  $m_1 = 0$  to  $D_1 - 1$  do
  for  $m_2 = 0$  to  $D_2 - 1$  do
     $\vdots$ 
    for  $m_n = 0$  to  $D_n - 1$  do
      for  $k = 1$  to  $H$  do
         $a_k^m = \sum_{i=1}^I x_i^m w_{ik}$ 
      end
      for  $d = 1$  to  $n$  do
        if  $m_d > 0$  then
           $a_k^m + = \sum_{j=1}^H h_j^{(m_1, \dots, m_{d-1}, \dots, m_n)} w_{jk}^d$ 
           $h_k^m = \phi_k(a_k^m)$ 
        end
      end
    end
  end
end

```

Algorithm 2: MDRNN backward pass

```

for  $m_1 = D_1 - 1$  to  $0$  do
  for  $m_2 = D_2 - 1$  to  $0$  do
     $\vdots$ 
    for  $m_n = D_n - 1$  to  $0$  do
      for  $k = 1$  to  $H$  do
         $e_k^m \leftarrow \sum_{j=1}^K \alpha_j^{\hat{p}} w_{jk}$ 
        if  $m_d < D_d - 1$  then
           $e_k^m = \sum_{j=1}^H \hat{h}_j^{(m_1, \dots, m_{d+1}, \dots, m_n)} w_{jk}^d$ 
           $h_k^{\hat{m}} \leftarrow \phi_k'(e_k^m)$ 
        end
      end
    end
  end
end

```

Here, the derivatives of the cost function with respect to the activations are $\alpha_j^{\hat{p}}$ and $h_k^{\hat{p}}$ for j th input and the k th hidden unit at point p .

As forward or backward pass is required, the training intricacy of MDRNN is linear with respect to the number of data points and network weights.

The equations for forward pass are:

$$a_k^m = \sum_{i=1}^I x_i^m w_{ik} + \sum_{\substack{d=1 \\ m_d > 0}}^n \sum_{j=1}^H h_j^{(m_1, \dots, m_{d-1}, \dots, m_n)} w_{jk}^d \quad (9)$$

and:

$$h_k^m = \phi_k(a_k^m) \quad (10)$$

The equation for backward pass is:

$$\hat{h}_k^m = \phi_k'(a_k^m) \left(\sum_{j=1}^K \alpha_j^{\hat{m}w_{jk}} + \sum_{d=1}^n \sum_{j=1}^H h_k^{(\hat{m}_1, \dots, \hat{m}_{d+1}, \dots, \hat{m}_n)} w_{jk}^d \right) \quad (11)$$

$m_d < D_d - 1$

The above equations will be analogous to RNN for $n = 1$.

4 Proposed Method

In this work, MDRNN dimension is taken as two and hence two numbers of RNNs have been combined and replaced by one MDRNN. One dimensional RNNs of Left-to-Right and Right-to-Left are replaced by one MDRNN. Similarly, one dimensional RNNs of Top-to-Bottom and Bottom-to-Top are replaced by another MDRNN as shown in Fig. 3. MDRNN will learn spatial dependencies between the middle-level visual patterns and the fully connected layers will learn global image representation.

Numerous convolutional layers will extract more abstract and robust patterns. With back propagation in the network, the global representations from the output will be propagated back to MDRNN to improve the spatial dependency encoding. Further, CNNs will be fed to learn both middle and low-level features. The addition of the results of two MDRNNs will be processed at the output by fully connected layers.

The network with access to the surrounding context in all directions is generally preferred. The issue in multidirectional context for one dimensional RNNs can be resolved by introducing two separate hidden layers that process the input sequence in the forward and reverse directions. The two hidden layers are connected to a single output layer, thereby providing the network with access to both past and future contexts. This can be extended to n -dimensional data by using 2^n separate hidden layers. If the size of the hidden layers is held constant, the multi-directional MDRNN architecture scales as $O(2^n)$ for n -dimensional data. However, the computing power of the network will not get affected, as it depends on the overall number of weights, rather than the size of the hidden layers. This is because the data processing is shared between the layers. Hence, the $O(2^n)$ scaling factor can be neutralized by using smaller hidden layers for higher dimensions. Moreover, the complexity of a task and the number of weights are likely to be needed for it. Complexity does not necessarily increase with the dimensionality of the data.

For training the MDRNN, the loss function is chosen as cross-entropy as in CNN and RNN and mini-batch gradient descent is used to find the best parameters of the network. Training a neural network is to find the best

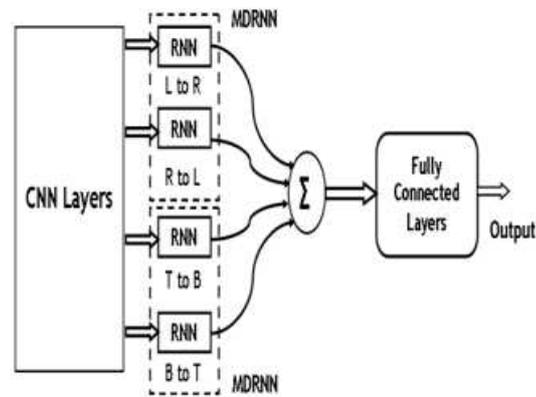


Fig. 3: Block diagram of MDRNN-CNN framework with two one dimensional RNNs

parameters (weights of the network) to minimize the loss function and classification task to measure the compatibility between a prediction (e.g., the class scores in classification) and the ground truth label.

The loss takes the form of an average over the losses for all training iterations as:

$$L = \frac{1}{N} \sum_{k=1}^N L_k \quad (12)$$

where N is the number of samples and L_k is the k th sample loss.

For the output layer with softmax activation, the cross-entropy loss is defined as:

$$L_k = -\log(p(y_k | x_k)) \quad (13)$$

This is a negative log-likelihood function computed for a training iteration.

5 Experimental Results and Discussion

In this work, two benchmark HSI datasets: Pavia University images and Salinas images are considered. The Reflective Optics System Imaging Spectrometer (ROSIS) sensor collected Pavia University images. It consists of 103 spectral bands, with 430 nm to 860 nm spectral range. The spatial resolution of the image is 1.3 m, and the total image size is 610×340 pixels. The Salinas images were acquired via the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS), and the image size is 512×217 , with spatial resolution of 3.7 m. It consists of 224 spectral bands. After removing 20 bands containing noise and water-absorption, 204 spectral bands are left for subsequent analysis. The ground-reference data for the Salinas images entails 16 classes. The false-color composite and ground-reference map of the Pavia University image with classes are shown in Fig. 4. The false-color composite and ground-reference map of the Salinas image with classes are shown in Fig. 5.

Table 1: Class codes for Pavia University and Salinas images

Pavia University Image		Salinas Image	
Class No.	Name	Class No.	Name
1	Asphalt	1	Broccoli_green_weeds_1
2	Meadow	2	Broccoli_green_weeds_2
3	Gravel	3	Fallow
4	Trees	4	Fallow_rough_plow
5	Painted Metal Sheets	5	Fallow_smooth
6	Bare Soil	6	Stubble
7	Bitumen	7	Celery
8	Self-Blocking Bricks	8	Grapes_untrained
9	Shadows	9	Soil_vinyard_develop
		10	Corn_senesced_green_weeds
		11	Lettuce_romaine_4wk
		12	Lettuce_romaine_5wk
		13	Lettuce_romaine_6wk
		14	Lettuce_romaine_7wk
		15	Vinyard_untrained
		16	Vinyard_vertical_trellis

Table 2: Number of Training and Testing Samples used for Pavia University

Class No.	Name	Training Samples	Test Samples
1	Asphalt	550	6080
2	Meadow	540	7110
3	Gravel	390	2705
4	Trees	540	4520
5	Painted Metal Sheets	250	2080
6	Bare Soil	530	4500
7	Bitumen	375	1955
8	Self-Blocking Bricks	510	3170
9	Shadows	245	1700

5.1 Distance measure

The distance measure used in this work is Spectral Angle Mapper (SAM) [25]. The SAM is the mapping of the spectral coincidence of test image spectra with reference spectra. The spectral similarity between the two spectra is defined as:

$$SAM = \cos^{-1} \left(\frac{I_t \cdot I_r}{\|I_t\| \cdot \|I_r\|} \right) \quad (14)$$

This can also be written as:

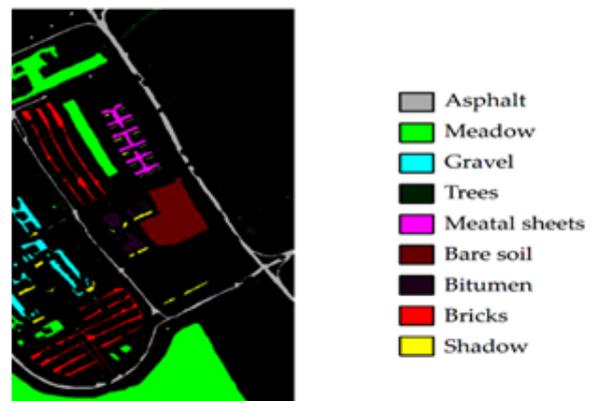
$$SAM = \cos^{-1} \left(\frac{\sum_{i=1}^N I_{ti} I_{ri}}{(\sum_{i=1}^N I_{ti}^2)^{1/2} (\sum_{i=1}^N I_{ri}^2)^{1/2}} \right) \quad (15)$$

where I_t is the test image vector, I_r is the reference image vector and N is the number of bands.

Table 1 lists the class codes for Pavia University and Salinas image datasets. Tables 2 and 3 provide corresponding training and test samples used in this work.



(a) False-color composite of Pavia University image



(b) Ground-Reference map of the Pavia University image with classes

Fig. 4

To assess the effectiveness of the proposed MDRNN, three algorithms SVM, CNN, and RCNN are taken as baseline algorithms. For SVM, the Radial Basis Function (RBF) is utilized as kernel function. For the CNN, two convolutional layers, two max pooling layers, and one Fully-Connected (FC) layer are considered. Different model structures are implemented based on different images.

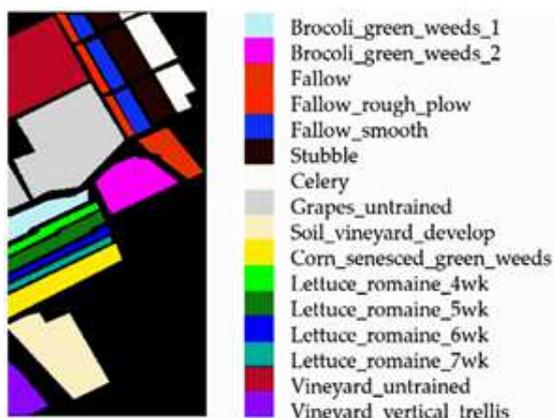
All ground-reference data for each image are randomly split into training and testing sample sets for evaluating the classification accuracy. Ten independent runs were performed on each data set. The results are averaged across the ten runs and tabulated.

Furthermore, it is observed that the improvement in accuracy in RCNN as compared to CNN reaches a maximum of 42% for some cases like class 6 but no improvement in the case of class 9 in Pavia University data. Conversely, the improvement is uniform for all classes in MDRNN compared to RCNN. Fig. 6 and 7 show that the classified result of MDRNN preserves all the relevant data of Ground-Truth.

Tables 4 and 5 provide the results found by the experiment, and Figs. 6 and 7 demonstrate the



(a) False-color composite of Salinas image



(b) Ground-Reference map of the Salinas image with classes

Fig. 5

Table 3: Number of Training and Testing Samples used for Salinas images

Class No.	Name	Training Samples	Test Samples
1	Brocoli_green_weeds_1	30	15
2	Brocoli_green_weeds_2	150	1200
3	Fallow	150	680
4	Fallow_rough_plow	100	130
5	Fallow_smooth	150	340
6	Stubble	150	580
7	Celery	20	10
8	Grapes_untrained	150	320
9	Soil_vinyard_develop	15	10
10	Corn_senesced_green_weeds	150	810
11	Lettuce_romaine_4wk	150	1230
12	Lettuce_romaine_5wk	150	440
13	Lettuce_romaine_6wk	150	75
14	Lettuce_romaine_7wk	150	1100
15	Vinyard_untrained	50	340
16	Vinyard_vertical_trellis	50	45

Table 4: Classification Accuracy for the Pavia University image in %

Class No.	SVM	CNN	RCNN	MDRNN
1	97.52 ± 0.21	96.53 ± 0.57	98.78 ± 0.39	99.02 ± 0.23
2	95.77 ± 0.30	94.78 ± 1.55	98.84 ± 0.29	99.17 ± 0.31
3	65.57 ± 3.41	68.93 ± 3.65	89.22 ± 2.63	91.54 ± 3.00
4	71.27 ± 7.38	76.62 ± 7.92	94.70 ± 1.71	95.92 ± 1.29
5	95.50 ± 1.55	98.50 ± 0.86	99.42 ± 0.64	99.71 ± 0.53
6	59.51 ± 6.74	63.62 ± 11.03	90.65 ± 3.92	91.83 ± 4.04
7	52.10 ± 0.93	66.87 ± 4.86	88.47 ± 6.60	90.19 ± 5.61
8	84.27 ± 1.13	83.54 ± 1.73	93.27 ± 1.28	94.72 ± 1.55
9	99.92 ± 0.11	99.65 ± 0.31	98.49 ± 1.75	99.88 ± 0.42

Table 5: Classification Accuracy for the Salinas image in %

Class No.	SVM	CNN	RCNN	MDRNN
1	96.84 ± 1.18	99.12 ± 1.52	99.86 ± 0.22	99.91 ± 0.19
2	98.79 ± 0.13	98.79 ± 0.69	99.34 ± 0.51	99.74 ± 0.15
3	85.11 ± 1.38	95.53 ± 1.32	96.24 ± 1.10	97.93 ± 0.95
4	97.44 ± 0.18	97.94 ± 0.67	97.90 ± 1.70	98.32 ± 1.01
5	95.03 ± 0.85	97.20 ± 2.76	98.83 ± 0.68	99.65 ± 0.92
6	99.79 ± 0.11	99.77 ± 0.21	99.71 ± 0.33	99.80 ± 0.24
7	98.63 ± 0.44	99.35 ± 0.62	99.38 ± 0.51	99.61 ± 0.32
8	76.70 ± 1.31	83.99 ± 4.02	85.66 ± 3.00	87.11 ± 2.78
9	99.12 ± 0.04	99.02 ± 0.29	99.43 ± 0.27	99.67 ± 0.19
10	81.91 ± 1.58	85.89 ± 2.09	91.00 ± 2.13	92.24 ± 1.99
11	69.51 ± 1.00	82.23 ± 6.71	83.49 ± 4.40	85.38 ± 3.79
12	93.33 ± 0.34	96.82 ± 1.05	98.26 ± 1.67	99.21 ± 0.93
13	92.67 ± 0.65	94.15 ± 2.62	97.33 ± 1.98	98.58 ± 1.54
14	89.68 ± 2.19	89.80 ± 3.78	90.75 ± 3.77	90.75 ± 3.77
15	56.78 ± 2.39	59.34 ± 12.38	69.85 ± 3.21	72.04 ± 4.02
16	95.59 ± 1.18	98.60 ± 0.44	96.19 ± 3.79	97.93 ± 2.65

classification maps on the Pavia University dataset and the Salinas dataset. The accuracies given in Tables 4 and 5 are Overall Accuracies (OA) along with the standard deviation. The experiment is implemented with Python 3.6.

MDRNN outperforms SVM, CNN and RCNN for both data sets. In addition, it is inferred that MDRNN gives invariably better classification accuracy across all classes for both data sets. Thus, it is reasonable to specify that MDRNN is a better choice for HSI classification.

6 Conclusion

In this paper, a novel MDRNN HSI classification framework is proposed, where the sequential feature from a single HSI is extracted and classified accurately. The accuracy attains a maximum value of 99.91% for class 1 of Salinas image data set with MDRNN. The proposed algorithm provides high accuracy classification compared

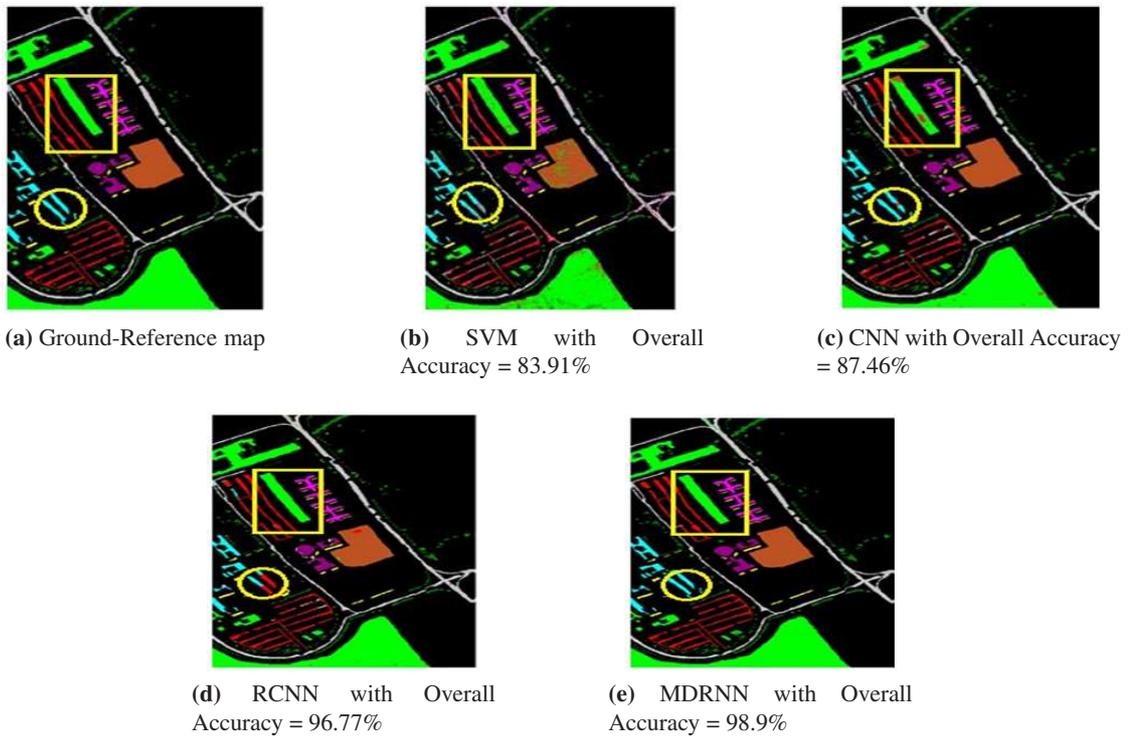


Fig. 6: Classification maps for the Pavia University image

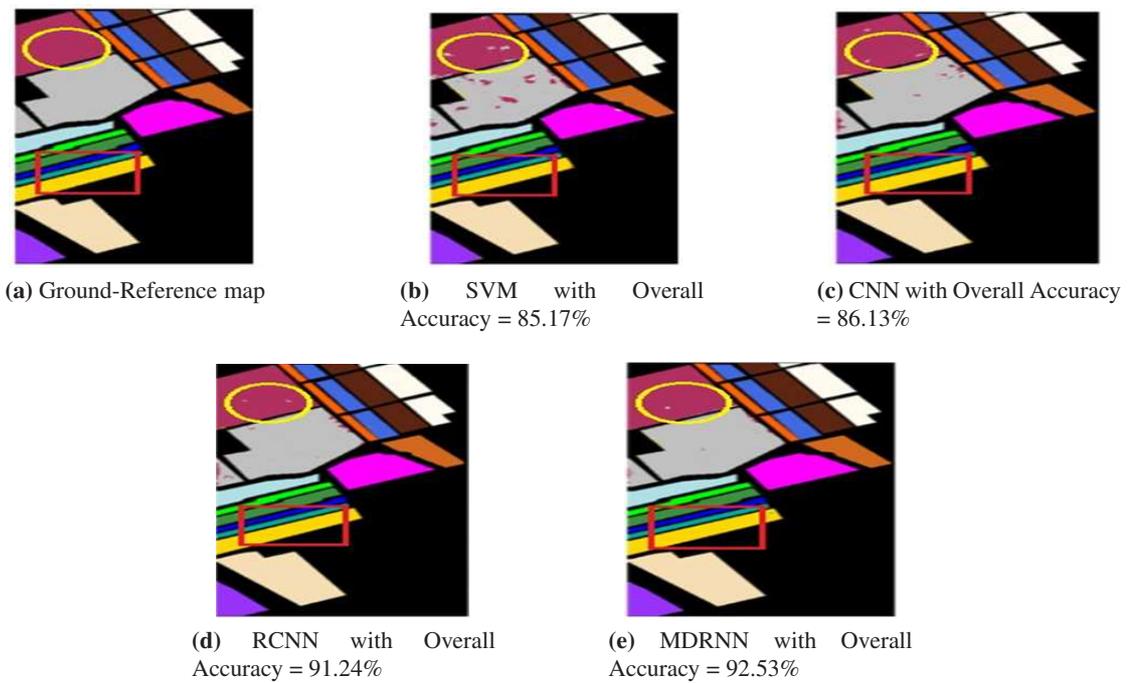


Fig. 7: Classification maps for the Salinas image

to existing algorithms like SVM, CNN and RCNN. The resultant improvement of the overall accuracy for the Pavia University image and the Salinas image is 2.2% and 1.3% respectively compared to RCNN algorithm. Hence, it is inferred that the proposed method is performing better on both image datasets.

The proposed method uses similarity measurements with SAM. The methods are yet to take up temporal context. The requirement of ground truth data is a challenging issue in handling hyperspectral datasets with deep learning methods. To prevent over-fitting, data augmentation is employed in deep learning methods. The spectral characteristics can be utilized to get better classification performance.

In summary, the developed deep learning method high accuracy classification HSI classification, with approximately 98% accuracy, approximately 6% improvement over the existing deep learning methods, and approximately 12% higher than the traditional method. MDRNN is an accurate classification technique that makes a significant contribution in the field of HSI classification and has great potential for a wide range of remote sensing applications.

References

- [1] D. Haboudane, J. R. Miller, E. Pattey, P. J. Zarco-Tejada, I. B. Strachan, Hyperspectral vegetation indices and Noel algorithms for predicting green LAI of crop canopies: Modeling and validation in the context of precision agriculture, *Remote Sens. Environ.*, **90**(3), 337–352 (2004).
- [2] G. Camps-Valls, D. Tuia, L. Bruzzone, J. A. Benediktsson, Advances in hyperspectral image classification: Earth monitoring with statistical learning methods, *IEEE Signal Processing Magazine*, **31**(1), 45–54 (2014).
- [3] Freek D Van der Meer, Harald MA Van der Wer, Frank JA van Ruitenbeek, Chris A Hecker, Wim H Bakker, Marleen F Nomen, Mark van der Meijde, E John M Carranza, J Boudewijn de Smeth, and Tsehaie Woldai, Multi-and hyperspectral geologic remote sensing: A review, *International Journal of Applied Earth Observations and Geoinformation*, **14**(1), 112–128 (2012).
- [4] P. WT Yuen and M. Richardson, An introduction to hyperspectral imaging and its a publication for security, surveillance and target acquisition, *The Imaging Science Journal*, **58**(7), 241–253 (2010).
- [5] Q. Wang, Z. Yuan, Q. Du, X. Li, GETNET: A General End-to-End 2-D CNN Framework for Hyperspectral Image Change Detection, *IEEE Trans. Geosci. Remote Sens.*, **57**(1), 3–13 (2018).
- [6] Hyungtae Lee and Heesung Kwon, Going deeper with contextual CNN for hyperspectral image classification *IEEE Transactions on Image Processing*, **26**(10), 4843–4855 (2017).
- [7] LichaoMou, Pedram Ghamisi, and Xiao Xiang Zhu, Deep recurrent neural networks for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, **55**(7), 3639–3655 (2017).
- [8] H. Wu and S. Prasad, Convolutional recurrent neural networks for hyperspectral data classification, *Remote Sens.*, **9**(3), 298 (2017).
- [9] G. Cheng, J. Han, X. Lu, Remote sensing image scene classification: benchmark and state of the art, *Proc. IEEE*, 1–19 (2017)
- [10] J. M. Haut, M. Paoletti, J. Plaza and A. Plaza, Cloud implementation of the k means algorithm for hyperspectral image analysis, *J. Supercomputing*, **73**(1), 514–529 (2017).
- [11] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias and X. Li, Spectral-spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization, *IEEE Trans. Geosci. Remote Sens.*, **52**(10), 6298–6314 (2014).
- [12] B. Scholkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA (2001).
- [13] D. Böhning, Multi Noial logistic regression Algorithm, *Ann. Inst. Stat. Math.*, **44**(1), 197–200 (1992).
- [14] J. Li, J. M. Bioucas-Dias and A. Plaza, Semi supervised hyperspectral image segmentation using multi-nomial logistic regression with active learning, *IEEE Trans. Geosci. Remote Sens.*, **48**(1), 4085–4098 (2010).
- [15] H. Yang, A back-propagation neural network for mineralogical mapping from AVIRIS data, *Int. J. Remote Sens.*, **20**(1), 97–110 (1999).
- [16] Y. Chen, Z. Lin, X. Zhao, G. Wang and Y. Gu, Deep learning-based classification of hyperspectral data, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **7**(6), 2094–2107 (2014).
- [17] Y. Chen, X. Zhao and X. Jia, Spectral-spatial classification of hyperspectral data based on deep belief network, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **8**(6), 2381–2392 (2015).
- [18] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, *IEEE Transactions on Geoscience and Remote Sensing*, **54**(10), 6232–6251 (2016).
- [19] A. J. Guo and F. Zhu, Spectral-spatial feature extraction and classification by ANN supervised with center loss in hyperspectral imagery, preprint arXiv:1711.07141, 2017.
- [20] L. Mou, P. Ghamisi and X. X. Zhu, Deep recurrent neural networks for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, **55**(7), 3639–3655 (2017).
- [21] Q. Liu, F. Zhou, R. Hang and X. Yuan, Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification, *Remote Sensing*, **9**, 1330 (2017).
- [22] ZhenZuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang and Yushi Chen, Convolutional Recurrent Neural Networks: Learning Spatial Dependencies for Image Representation, *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2015).
- [23] P. Baldi and G. Pollastri, The principle design of large-scale recursive neural network architectures–DAGRNNs and the protein structure prediction problem, *Journal on Mach. Learn. Res.*, **4**, 575–602 (2003).
- [24] J. Graves and Schmidhuber, Offline Handwriting Recognition with Multidimensional Recurrent Neural

Networks, *Advances in Neural Information Processing Systems*, **22**, 545–552 (2009)

- [25] F. A. Kruse, A. B. Lefkoff, J. B. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and A. F. H. Goetz, *The Spectral Image Processing System (SIPS)-Interactive Visualization and Analysis of Imaging spectrometer Data, Remote Sensing of Environment*, **44**(3), 145–163 (1993).



K. Thilagavathi

has 17 years of teaching experience at college level of both UG and PG. She is currently working as Assistant Professor in the Department of Electronics and Communication Engineering at Kumaraguru

College of Technology, Coimbatore. She has published 16 papers in International journals and over 27 papers in National and International conferences. Her area of interests includes Hyperspectral Remote Sensing, Communication Systems and Microwave Engineering.



A. Vasuki has more than 26 years of experience in teaching, research and administration. She is currently a Professor of Mechatronics Engineering at Kumaraguru College of Technology, Coimbatore. She has published 3 chapters in a

book, 32 papers in International journals and over 60 papers in National and International conferences. She has completed three funded research projects under RPS, AICTE and RESPOND, ISRO. She is currently guiding several research scholars for the doctoral programme. Her areas of interest are Signal / Image Processing and Communications.