

# Multi-layer Extreme Learning Machine-based Autoencoder for Hyperspectral Image Classification

Muhammad Ahmad<sup>1,2,\*</sup>, Adil Mehmood Khan<sup>1</sup>, Manuel Mazzara<sup>1</sup> and Salvatore Distefano<sup>2</sup>

<sup>1</sup>*Innopolis University, Innopolis, Russia*

<sup>2</sup>*University of Messina, Messina, Italy*

**Keywords:** Extreme Learning Machine (ELM), Deep Neural Networks (DNN), Auto Encoder (AE), Hyperspectral Image Classification.

**Abstract:** Hyperspectral imaging (HSI) has attracted the formidable interest of the scientific community and has been applied to an increasing number of real-life applications to automatically extract the meaningful information from the corresponding high dimensional datasets. However, traditional autoencoders (AE) and restricted Boltzmann machines are computationally expensive and do not perform well due to the Hughes phenomenon which is observed in HSI since the ratio of the labeled training pixels on the number of bands is usually quite small. To overcome such problems, this paper exploits a multi-layer extreme learning machine-based autoencoder (MLELM-AE) for HSI classification. MLELM-AE learns feature representations by adopting a singular value decomposition and is used as basic building block for learning machine-based autoencoder (MLELM-AE). MLELM-AE method not only maintains the fast speed of traditional ELM but also greatly improves the performance of HSI classification. The experimental results demonstrate the effectiveness of MLELM-AE on several well-known HSI dataset.

## 1 INTRODUCTION

Hyperspectral images (HSI) provides a unique way for characterizing objects of interest in geographical scenes with very rich spatial-spectral information contained in a 3-D hypercube (Ahmad et al., 2016). However, classification of such high dimensional hyperspectral data is still a challenging task, especially in the case the ratio between the number of available labeled training samples and the number of spectral dimensions (usually large) is small, which is commonly known as Hughes phenomenon (Hughes, 1968).

To cope with the issues due to the high number of dimensions, a number of feature extraction, selection, and classification methods have been proposed in the recent years (Ren et al., 2014; Ahmad et al., 2011; Liu et al., 2018). These methods have yielded quite good outcomes. However, their performance can be further improved by addressing two main issues: 1) inaccurate classification in the case of the Hughes phenomenon (Ahmad et al., 2018); 2) comparatively low efficiency for processing high dimensional HSI data (Ahmad et al., 2017a).

Extreme learning machine (ELM), as a single hid-

den layer feed-forward neural network, is an effective and fast machine learning method and has received a remarkable attention due to its high generalization performance (Ding et al., 2015). In ELM, the hidden layer parameters need to not be tuned once the number of hidden layer nodes is learned. Moreover, the bias and weights between the hidden and input layers are randomly assigned without taking into account the training samples and applications (Zhou et al., 2015).

Due to its generalization capabilities, ELM has been extensively studied for HSI classification problems, for instance in (Arguello and Heras, 2015; Shen et al., 2016), extended morphological profiles and bilateral filtering based methods were used for feature extraction and ELM was used as a base classifier. In (Chen et al., 2014; Dora et al., 2014), Gabor filter and watershed-based methods were employed for feature extraction and ELM was used as a final classification method. Regardless of computational complexity and other issues, these methods have achieved a remarkable performance for HSI classification. However, these methods ignore one very important aspect in ELM: the randomly rendered input bias and weights may cause ill-posed problems. Based on this phenomenon, to handle such a pro-

blem effectively and efficiently we intend to exploit the multi-layer extreme learning machine-based autoencoder (MLELM-AE) method for HSI classification when we do not need to extract the features explicitly, as mentioned in (Kasun et al., 2013), for digit classification problems. To the best of our knowledge, it is a first of its kind of work for HSI classification. A similar criterion has been explored in the past (Kasun et al., 2013). However, in our work, instead of using the pipeline for traditional image classification or recognition, we implemented and tested it on hyperspectral image classification and segmentation problem which is more complexed than the traditional image classification.

The remainder of the paper is structured as follows. Section 2 presents the theoretical aspects of extreme learning machine pipeline followed by a theoretical explanation of the extreme learning machine learning based autoencoder. Section 3 discusses experimental setups and metrics. Section 4 discussed the dataset, settings, and results. Finally, Section 5 summarizes the contributions and future research directions.

## 2 EXTREME LEARNING MACHINE

In ELM, the bias and weight vectors between the hidden and input layer are randomly assigned, while the net values are obtained by the learning process. Once the initial values are preserved, the hidden layer output matrix persist unaltered in the learning process.

Let as assume,  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N) \in \mathbb{R}^{d \times N}$  be the training data which has  $N$  number of pixels and each pixel has  $d$ -dimensional feature. Let  $\mathbf{Y} = (y_1, y_2, y_3, \dots, y_M) \in \mathbb{R}^{M \times N}$  be a matrix representing the class labels of the training samples in which  $M$  is the number of classes in HSI data. Thus, the ELM model with  $L$  hidden neurons and the activation function  $H(x)$  can be expressed as;

$$\sum_{j=1}^L \beta_j \mathbf{H}(W_j^T \mathbf{x}_i + b_j) = y_i; i = 1, 2, 3, \dots, N \quad (1)$$

where  $\mathbf{H}(W_j^T \mathbf{x}_i + b_j)$  represents the output of the  $j^{\text{th}}$  hidden neuron with respect to the input  $\mathbf{x}_i$  and  $\beta_j$ ,  $W_j$ , and  $b_j$  represents the weight vector between hidden layer and output layer, and weight and bias between hidden and input layer, respectively. The above expression can simply be written as;

$$\mathbf{H}^T \beta = \mathbf{Y}^T \quad (2)$$

where

$$\beta = [\beta_1, \beta_2, \beta_3, \dots, \beta_M]_{L \times M} \quad (3)$$

$$\mathbf{H} = [\mathbf{H}(\mathbf{x}_1), \mathbf{H}(\mathbf{x}_2), \mathbf{H}(\mathbf{x}_3), \dots, \mathbf{H}(\mathbf{x}_n)]_{L \times N} \quad (4)$$

and

$$\mathbf{H}(\mathbf{x}_i) = [\mathbf{H}_1(\mathbf{x}_i), \mathbf{H}_2(\mathbf{x}_i), \mathbf{H}_3(\mathbf{x}_i), \dots, \mathbf{H}_L(\mathbf{x}_i)]_{L \times 1}^T \quad (5)$$

Finally,  $\beta$  can be computed as;

$$\beta \approx (\mathbf{H}^T)^{\dagger} \mathbf{Y}^T \quad (6)$$

where  $(\cdot)^{\dagger}$  is the Moore Penrose generalized inverse of a matrix.

The main goal of multi-layer extreme learning machine-based autoencoder is to learn a useful feature representation in three different folds similar to traditional autoencoders (Ahmad et al., 2019). Namely, compressed representation - manifest input features form high dimensional hyperspectral space to a lower dimensional feature space, sparse representation - low dimensional input feature space to higher dimensional hyperspectral feature space, and finally, equal input/output dimensional representation - interpret input space dimensions equal to feature space dimension.

According to (Kasun et al., 2013; Huang et al., 2006), extreme learning machine is a universal approximator, therefore, MLELM-AE is also a universal approximator. In MLELM-AE the orthogonal random biases and weights of the hidden nodes undertaken the input samples to equal dimensional space as shown in (Kasun et al., 2013; Huang et al., 2006; Johnson and Lindenstrauss, 1984) and in below equation similar to the equation (1),

$$\mathbf{h} \approx g(\mathbf{a} \times \mathbf{x} + b) \quad (7)$$

where  $\mathbf{a}^T \times \mathbf{a} = \mathbf{I}$  in which  $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L]$ , and  $b^T \times b = 1$  in which  $b = [b_1, b_2, \dots, b_L]$  are the orthogonal random weights and bias between input and hidden nodes, respectively. Therefore, as shown in (Kasun et al., 2013; Huang et al., 2006), the output weights for compressed and sparse MLELM-AE representation can be obtained by incorporating the regularization term to enhance the generalization performance and robustness

$$\beta \approx \left( \frac{\mathbf{I}}{C} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{X} \quad (8)$$

where  $C$  is the regularization term,  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N]$  are input and output data samples, and  $\mathbf{H} = [h_1, h_2, h_3, \dots, h_N]$  are the hidden layer outputs of MLELM-AE. The output weights can also be computed as;

$$\beta \approx \mathbf{H}^{-1} \mathbf{X} \quad (9)$$

where  $\beta^T \beta = \mathbf{I}$  to make the input and output equal. Therefore, the singular value decomposition of regularized output weights for compressed and sparse MLELM-AE representation can be computed as in (Kasun et al., 2013), i.e.

$$\mathbf{H}\beta \approx \sum_{i=1}^N \mathbf{u}_i \frac{v_i^2}{v_i^2 + C} \mathbf{u}_i^T \mathbf{X} \quad (10)$$

where  $v$  represents the singular values of  $\mathbf{H}$  and  $\mathbf{u}$  represents the eigenvectors of  $\mathbf{H}\mathbf{H}^T$ . Since,  $\mathbf{H}$  is the projected feature space squashed via a linear or nonlinear (sigmoid or any appropriate) activation function, we speculate that the output weights  $\beta$  will be learning to represent the features of the input space via singular value decomposition.

Furthermore, if the number of hidden nodes  $L^k$  in  $k^{\text{th}}$  hidden layer is equal to the number of hidden nodes  $L^{k-1}$  in the  $(k-1)^{\text{th}}$  hidden layer in which  $g$  is chosen as linear activation function other way around  $g$  will be chosen as nonlinear piece-wise activation function. This way

$$\mathbf{H}^k \approx g((\beta^k)^T \mathbf{H}^{k-1}) \quad (11)$$

where  $\mathbf{H}^k$  is the  $k^{\text{th}}$  hidden layer output matrix. For better intuition, the input after  $\mathbf{x}$  can be identified as the  $0^{\text{th}}$  hidden layer where  $k=0$ . Finally, the output of the connections among the last hidden layer and the output node  $t$  is over-analytically computed by employing regularized least squares, where  $t$  is the output data.

### 3 EXPERIMENTAL METRICS

In this section, the performance of MLELM-AE is evaluated using seven different well-known publicly available AVIRIS, ROSIS, and NASA EO-1 satellite Hyperion sensor based hyperspectral datasets. More information about these datasets can be found in (Liu et al., 2018; Ahmad et al., 2018; Ahmad et al., 2017b; Li et al., 2013; He et al., 2018; Datasets, ).

Confusion matrix is generally used to evaluate the performance of HSI classification in terms of overall, average accuracy, and kappa  $\kappa$  coefficient. In this work, the overall accuracy for hyperspectral image classification is computed by the following formula:

$$OA = \frac{\sum_{i=1}^N x_{ii}}{\sum_{j=1}^M \sum_{i=1}^N x_{ij}} \quad (12)$$

From the above equation, it can be seen that the magnitude of the overall accuracy is only affected by the diagonal elements. It is more likely affected by

classes that contain more elements so it is not sufficient to comprehensively evaluate the classification accuracy of all classes. A more comprehensive index of classification accuracy evaluation is the  $\kappa$  coefficient utilizing all samples of the confusion matrix thus reflecting the consistency between classification results and ground truth. The  $\kappa$  coefficient is evaluated by the formula (Ahmad et al., 2017b):

$$\kappa = \frac{N \sum_i a_i - \sum_i b_i d_i}{N^2 - \sum_i b_i d_i} \quad (13)$$

where  $N$  is the total number of samples (pixels in HSI cube),  $a_i$  is the number of correctly predicted samples in the given class,  $\sum_i a_i$  is the sum of the number of correctly predicted samples,  $b_i$  is the actual number of samples belonging to the given class and  $d_i$  is the number of samples that have been correctly predicted into the given class (Ahmad et al., 2018).

Furthermore, to evaluate the significance of MLELM-AE, several statistical tests are conducted e.g, F1-score, precision, and recall rate. The precision maps the ratio of correctly identified positive samples to the total predicted positive samples. High precision value indicates lesser false positive rates referring to the model ability to correctly identify the true positive samples. Whereas, recall accounts the ratio of correctly predicted positive samples from the entire positive samples as true. As similar to precision, the higher recall rate the better the model is.

Likewise, F1-score is a weighted average of precision and recall rates. Therefore, F1-score takes both false negatives and false positives into account. F1-score is more useful than the other accuracy measures, but intuitively not as easy to understand as accuracy, particularly when we have unbalanced class distribution. Several accuracy measures works well if false negatives and positives have similar cost, if in case these are different, then better to consider both precision and recall rate to evaluate the model.

In this section, we will also evaluate the relevant tuning parameters which include the number of neurons in the hidden layers, the total number of layers, and the appropriate value for regularization term  $C$ . In our experiments, the regularization term is automatically tuned by the 5-fold-cross-validation process. The number of hidden layer neurons is systematically set from the range  $[Total\ Number\ of\ Training\ Samples - Total\ Number\ of\ Testing\ Samples]$ , and the number of layers is heuristically set in the range  $[1 - 5]$  for cross-validation process to find the optimum value of regularization term from the interval  $[1e^{-1} - 1e^{14}]$ .

## 4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we will briefly discuss the experimental results acquired by MLELM-AE pipeline on seven different hyperspectral datasets. Prior to the experiments, we performed the necessary normalization between  $[0 - 1]$ . All the experiments have been carried out on a cluster using MATLAB (2017a) on Intel Core (TM) i7-7700K CPU 2.40GHz, 1962 MHz, Ubuntu 16.01.5 LTS, Cude compilation tools, release 7.5, V7.5.17 with 65GB RAM.

The presented experiments shows the accuracy analysis in terms of overall accuracy, average accuracy, and  $\kappa$  coefficient. Figures 1-7 shows the ground truth maps for original test samples along with the prediction of these samples in geographical maps.

Furthermore, these Figures also presented the average, overall, and  $\kappa$  accuracy in multiclass form along with the mean squared error of MLELM-AE model in 10-fold-cross-validation along with the training and testing time for each dataset. The training and test time is significantly less than the traditional back propagation based deep neural networks. Furthermore, the plots shows higher generalization performance with less amount of training samples.

To highlight the class-based classification results, Tables 1-7 report the  $\kappa$  coefficient for each individual class, providing insights on the number of training versus estimated labels used in our experiments and thus demonstrating clear advantages of using limited samples for the learning MLELM-AE model. In most cases, the proposed pipeline outperforms existing solutions.

### Experiments with Salinas Dataset

The Salinas dataset consists of 224 spectral bands with a high spatial resolution of 3.7 m. Salinas full scene was collected by AVIRIS sensor over Salinas Valley California.

In Salinas scene some bands were water absorption and removed prior to the analyses. The removed bands are 108 – 112, 154 – 167 and 224. The full Salinas scene is covered with  $512 \times 217$  pixels per band and contains vegetables, bare, soils and vineyard field. Salinas ground truth contains 16 classes.

A sub-scene of Salinas dataset named Salinas-A consists of  $86 \times 83$  samples per band and 6 classes. The Salinas-A samples are located in the full Salinas scene at 591 – 676 and 158 – 240. Dataset files and description can be obtained from (Datasets, ).

The experimental results are shown in Tables 1 and 2 and Figures 1 and 2. From results, it can be clearly

seen that MLELM-AE pipeline greatly improved the classification accuracies for Salinas and Salinas-A datasets with acceptable generalization performance. Furthermore, the detailed accuracy and time taken to train and test the model is provided in the caption respective Figures. In all these experiments the training size is set as 1% samples from Salinas and Salinas-A datasets, respectively.

Table 1: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Salinas-A Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Broccoli Green Weeds 1	(8, 375)	0.9316 $\pm$ 0.0516	0.9948	0.9999	0.9973
Corn Senesced Green Weeds	(27, 1289)	0.9695 $\pm$ 0.0144	0.9795	0.9999	0.9896
Lettuce Romaine 4wk	(13, 591)	0.9773 $\pm$ 0.0212	0.9934	0.9772	0.9852
Lettuce Romaine 5wk	(31, 1464)	0.9975 $\pm$ 0.0022	0.9999	0.9973	0.9987
Lettuce Romaine 6wk	(14, 647)	0.9947 $\pm$ 0.0026	0.9969	0.9763	0.9865
Lettuce Romaine 7wk	(16, 767)	0.9757 $\pm$ 0.0125	0.9796	0.9783	0.9789

Table 2: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Salinas Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Broccoli Green Weeds 1	(61, 2009)	0.9982 $\pm$ 0.0003	0.9979	1.0000	0.9989
Broccoli Green Weeds 2	(112, 3726)	0.9968 $\pm$ 0.0004	0.9958	0.9991	0.9975
Fallow	(60, 1976)	0.8404 $\pm$ 0.0232	0.8721	0.9631	0.9153
Fallow Rough Plow	(42, 1394)	0.9835 $\pm$ 0.0033	0.9859	0.9925	0.9892
Fallow Smooth	(81, 2678)	0.9883 $\pm$ 0.0029	0.9795	0.9002	0.9382
Stubble	(119, 3959)	0.9971 $\pm$ 0.0005	0.9963	0.9994	0.9979
Celery	(108, 3579)	0.9963 $\pm$ 0.0007	0.9959	0.9945	0.9952
Grapes Untrained	(339, 11271)	0.8891 $\pm$ 0.0068	0.8795	0.7884	0.8314
Soil Vineyard Develop	(187, 6203)	0.9909 $\pm$ 0.0026	0.9960	0.9848	0.9904
Corn Senesced Green Weeds	(99, 3278)	0.9348 $\pm$ 0.0086	0.9465	0.9552	0.9508
Lettuce Romaine 4wk	(33, 1068)	0.9539 $\pm$ 0.0081	0.9748	0.9465	0.9605
Lettuce Romaine 5wk	(58, 1927)	0.9999 $\pm$ 0.0001	0.9994	0.9623	0.9806
Lettuce Romaine 6wk	(28, 916)	0.9785 $\pm$ 0.0018	0.9752	0.9569	0.9659
Lettuce Romaine 7wk	(33, 1070)	0.9284 $\pm$ 0.0083	0.9402	0.9587	0.9493
Vineyard Untrained	(219, 7268)	0.6238 $\pm$ 0.0105	0.6300	0.7857	0.6994
Vineyard Vertical Trellis	(55, 1807)	0.9840 $\pm$ 0.0020	0.9857	0.9976	0.9917

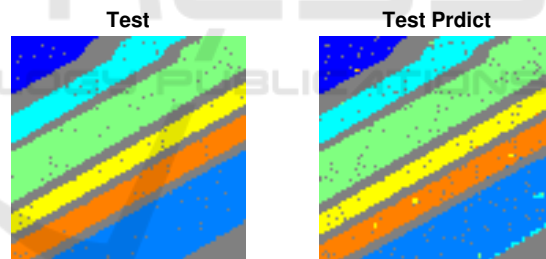


Figure 1: True Testing Maps and predicted Test Maps for Salinas-A dataset with 10-fold-cross-validation-based Average Accuracy =  $0.9744 \pm 0.0093$ , Overall Accuracy =  $0.9797 \pm 0.0059$ ,  $\kappa = 0.9746 \pm 0.0073$ , Mean-squared Error =  $0.2561 \pm 0.0478$ , and Training Time =  $0.0918 \pm 0.0067$  and Test Time =  $0.2539 \pm 0.0141$ .

### Experiments with Kennedy Space Center Dataset

The NASA AVIRIS instrument acquired data over the Kennedy Space Center (KSC) Florida on March 23, 1996. AVIRIS acquired data in 224 bands of 10 nm width with center wavelengths in the range 400 – 2500 nm, from an altitude of approximately 20 km with a spatial resolution of 18 m. After removing water absorption and low SNR bands, 176 bands were used for the analysis.

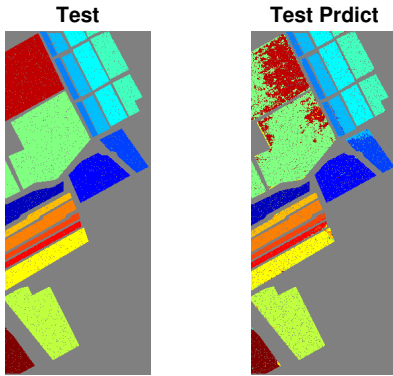


Figure 2: True Testing Maps and predicted Test M for Salinas dataset with 10-fold-cross-validation-based Average Accuracy =  $0.9428 \pm 0.0018$ , Overall Accuracy =  $0.9106 \pm 0.0013$ ,  $\kappa = 0.9002 \pm 0.0015$ , Mean-squared Error =  $1.9138 \pm 0.0087$ , and Training Time =  $54.2526 \pm 2.0612$  and Test Time =  $105.7566 \pm 3.6922$ .

Training data were selected using land cover maps derived from color infrared photography provided by the KSC and Landsat Thematic Mapper (TM) imagery. The vegetation classification scheme was developed by KSC personnel in an effort to define functional types that are discern-able at the spatial resolution of Landsat and this AVIRIS dataset.

Discrimination of land cover for this environment is difficult due to the similarity of spectral signatures for certain vegetation types. For classification purposes, 13 classes representing the various land cover types that occur in this environment were defined for the site. Dataset files and description can be taken from (Datasets, ). The experimental results are shown in Table 3 and Figure 3. From results, one can conclude that MLELM-AE greatly improved the classification accuracies for more complicated Kennedy space center AVIRIS sensor dataset with enhanced generalization capabilities. Moreover, the detailed accuracy and time taken to train and test the model is provided in the caption.

Table 3: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Kennedy Space Center Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Scrub	(229, 761)	$0.9895 \pm 0.0018$	0.9906	0.7877	0.8776
Willow Swamp	(73, 243)	$0.9541 \pm 0.0169$	0.8941	0.7958	0.8421
CP/Oak	(77, 252)	$0.9240 \pm 0.0225$	0.9162	0.6667	0.7718
CP hammock	(76, 256)	$0.4903 \pm 0.0388$	0.5000	0.8073	0.6175
Slash Pine	(49, 161)	$0.6384 \pm 0.0253$	0.6250	0.8434	0.7179
Oak/Broadleaf	(69, 229)	$0.2444 \pm 0.0192$	0.2875	0.7541	0.4163
Hardwood Swamp	(32, 105)	$0.4603 \pm 0.0658$	0.6575	0.7500	0.7007
Graminoid Marsh	(130, 431)	$0.8545 \pm 0.0133$	0.8638	0.9629	0.9107
Spartina Marsh	(156, 520)	$0.9841 \pm 0.0039$	0.9890	0.8933	0.9387
Cattail Marsh	(122, 404)	$0.9439 \pm 0.0089$	0.9362	0.9778	0.9565
Salt Marsh	(126, 419)	$0.9785 \pm 0.0044$	0.9659	0.9861	0.9759
Mud Flats	(151, 503)	$0.8773 \pm 0.0179$	0.9233	0.9207	0.9219
Water	(279, 927)	$0.9852 \pm 0.0043$	0.9784	0.9969	0.9875

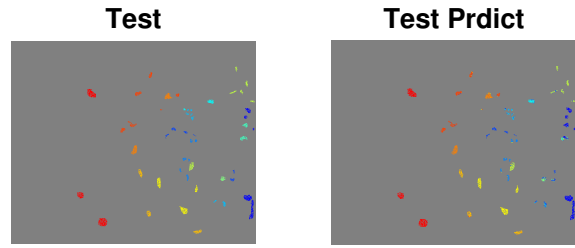


Figure 3: True Testing Maps and predicted Test Maps for Kennedy Space Center (KSC) dataset with 10-fold-cross-validation-based Average Accuracy =  $0.7942 \pm 0.0059$ , Overall Accuracy =  $0.8786 \pm 0.0034$ ,  $\kappa = 0.8642 \pm 0.0038$ , Mean-squared Error =  $1.4954 \pm 0.0374$ , and Training Time =  $14.2984 \pm 0.5296$  and Test Time =  $0.7258 \pm 0.0662$ .

## Experiments with Indian Pines Dataset

Indian Pines dataset is gathered by AVIRIS sensor over the Indian Pines test site in north-western Indiana and consists of  $145 \times 145$  pixels and 224 bands in the wavelength range  $0.4 - 2.5 \times 10^{-6}$  meters.

Indian Pines dataset contains 2/3 agriculture, and 1/3 forest or other natural perennial vegetation. There are two major dual lane highways, a rail line, as well as some low density housing, other build structures, and small roads. Since Indian Pines dataset was taken in June some of the crops present, corn, soybeans, are in early stages of growth with less than 5% coverage. The ground truth available is distinguished into sixteen classes not all mutually exclusive.

We have also reduced the number of bands to 200 by removing bands covering the region of water absorption. The removed bands are 104-108, 150-163, 220. Dataset files and description can be obtained from (Datasets, ). The experimental results are shown in Table 4 and Figure 4. From results, one can conclude that MLELM-AE greatly improved the classification accuracies with enhanced generalization capabilities. Furthermore, the detailed accuracy analysis and time taken to train and test the model is provided in the caption.

Table 4: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Indian Pines Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Alfalfa	(10, 46)	$0.4556 \pm 0.0736$	0.6944	0.8065	0.7464
Corn-notill	(286, 1428)	$0.8117 \pm 0.0070$	0.8284	0.7472	0.7857
Corn-mintill	(166, 830)	$0.6048 \pm 0.0215$	0.5768	0.7539	0.6536
Corn	(48, 237)	$0.3862 \pm 0.0385$	0.3492	0.7952	0.4853
Grass-pasture	(97, 483)	$0.8943 \pm 0.0159$	0.8834	0.9419	0.9118
Grass-trees	(146, 730)	$0.9829 \pm 0.0046$	0.9846	0.9055	0.9434
Grass-pasture-mowed	(6, 28)	$0.5409 \pm 0.0469$	0.5455	1.0000	0.7059
Hay-windrowed	(96, 478)	$0.9927 \pm 0.0038$	0.9921	0.9595	0.9756
Oats	(4, 20)	$0.1625 \pm 0.0688$	0.2500	0.8000	0.3809
Soybean-notill	(195, 972)	$0.6779 \pm 0.0137$	0.7091	0.7527	0.7303
Soybean-mintill	(491, 2455)	$0.8457 \pm 0.0053$	0.8432	0.7520	0.7951
Soybean-clean	(119, 593)	$0.7569 \pm 0.0172$	0.8101	0.8571	0.8329
Wheat	(41, 205)	$0.9866 \pm 0.0043$	0.9756	0.9639	0.9697
Woods	(253, 1265)	$0.9641 \pm 0.0047$	0.9664	0.9297	0.9477
Buildings-Grass-Trees-Drives	(78, 386)	$0.6042 \pm 0.0159$	0.6266	0.8143	0.7083
Stone-Steel-Towers	(19, 93)	$0.7284 \pm 0.0421$	0.8108	1.0000	0.8956

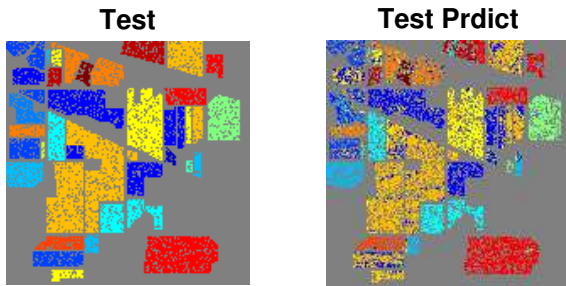


Figure 4: True Testing Maps and predicted Test Maps for Indian Pines dataset with 10-fold-cross-validation-based  $\kappa = 0.7839 \pm 0.0028$ , Average Accuracy =  $0.7122 \pm 0.0078$ , Overall Accuracy =  $0.8122 \pm 0.0024$ , Mean-squared Error =  $2.6312 \pm 0.0359$ , and Training Time =  $38.8711 \pm 0.7759$  and Test Time =  $3.2365 \pm 0.1330$ .

## Experiments with Pavia University and Pavia Center Datasets

The Pavia University (PU) dataset is acquired by the ROSIS optical sensor during a flight campaign over Pavia in northern Italy with geometric resolution of  $1.3m$ . PU data consists of 102 spectral bands with  $1096 \times 1096$  samples per band.

Some of the samples in PU dataset contains no information and have to be discarded prior to the analysis. PU scene ground-truths identified 9 classes. Dataset files and description can be obtained from (Datasets, ).

The experimental results are shown in Tables 5 and 6 and Figures 5 and 6. From results, one can conclude that MLELM-AE greatly improved the classification accuracies with enhanced generalization capabilities for more complicated ROSIS sensor based datasets. Evaluating ROSIS sensor datasets is more challenging classification problem dominated by complex urban classes and nested regions than AVIRIS. The detailed accuracy analysis in terms of average, overall, and kappa accuracies along with the time taken to train and test the model is provided in the caption.

Table 5: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Pavia University Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Asphalt	(1285, 6631)	$0.8105 \pm 0.0080$	0.8091	0.8916	0.8485
Meadows	(1980, 18649)	$0.9725 \pm 0.0016$	0.9694	0.8049	0.8795
Gravel	(93, 2099)	$0.6303 \pm 0.0133$	0.6406	0.7374	0.6856
Trees	(81, 3064)	$0.8115 \pm 0.0048$	0.8125	0.8199	0.8162
Painted metal sheets	(198, 1345)	$0.9947 \pm 0.0011$	0.9926	0.9917	0.9923
Bare Soil	(278, 5029)	$0.2468 \pm 0.0078$	0.2256	0.7964	0.3516
Bitumen	(219, 1330)	$0.6607 \pm 0.0116$	0.6859	0.8857	0.7731
Self-Blocking Bricks	(228, 3682)	$0.8296 \pm 0.0096$	0.8252	0.6421	0.7222
Shadows	(86, 947)	$0.8599 \pm 0.0140$	0.8885	0.9299	0.9088

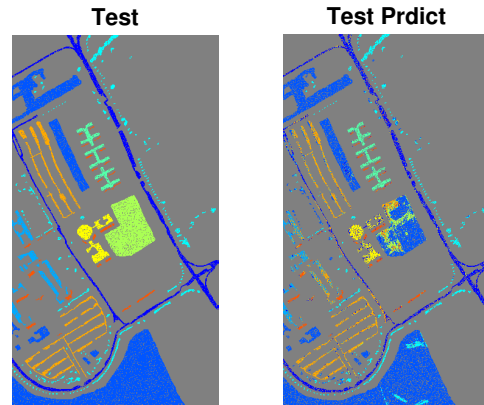


Figure 5: True Testing Maps and predicted Test Maps for Pavia University dataset with 10-fold-cross-validation-based Overall Accuracy =  $0.8099 \pm 0.0014$ , Average Accuracy =  $0.7574 \pm 0.0018$ ,  $\kappa = 0.7386 \pm 0.0019$ , Mean-squared Error =  $1.8703 \pm 0.0100$ , and Training Time =  $322.2472 \pm 2.6395$  and Test Time =  $70.9769 \pm 1.2438$ .

Table 6: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Pavia Center Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Water	(25, 824)	$0.9996 \pm 0.0001$	0.9997	0.9976	0.9986
Trees	(24, 820)	$0.9473 \pm 0.0055$	0.9499	0.8319	0.8870
Asphalt	(23, 816)	$0.5989 \pm 0.0211$	0.5849	0.7904	0.6723
Self Blocking Bricks	(21, 808)	$0.6389 \pm 0.0175$	0.6820	0.6612	0.6715
Bitumen	(21, 808)	$0.8887 \pm 0.0119$	0.8727	0.8287	0.8501
Tiles	(38, 1260)	$0.8173 \pm 0.0051$	0.8071	0.9364	0.8669
Shadows	(15, 476)	$0.8621 \pm 0.0091$	0.8877	0.9412	0.9136
Meadows	(25, 824)	$0.9957 \pm 0.0002$	0.9955	0.9778	0.9866
Bare Soil	(24, 820)	$0.8518 \pm 0.0123$	0.8739	0.8519	0.8628

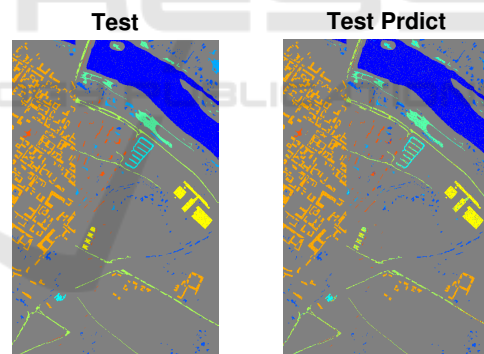


Figure 6: True Testing Maps and predicted Test Maps for Pavia Center dataset with 10-fold-cross-validation-based  $\kappa = 0.9360 \pm 0.0008$ , Average Accuracy =  $0.8445 \pm 0.0027$ , Overall Accuracy =  $0.9549 \pm 0.0006$ , Mean-squared Error =  $0.5067 \pm 0.0075$ , and Training Time =  $355.3899 \pm 3.6140$  and Test Time =  $410.2014 \pm 7.97897$ .

## Experiments with Botswana Dataset

The NASA EO-1 Satellite acquired a sequence of data over the Okavango Delta, Botswana in 2001-2004. The Hyperion sensor on EO-1 acquired data at  $30m$ , pixels resolution over a  $7.7km$  strip in 242 bands covering the  $400 - 2500nm$  portion of the spectrum in  $10nm$  windows.

Preprocessing of the data was performed by the UT center for space research to mitigate the effects of bad detectors, inter-detector mis-calibration, and intermittent anomalies. Uncalibrated and noisy bands that cover water absorption features were removed, and the remaining 145 bands were included as candidate features.

The removed features are 10-55, 82-97, 102-119, 134-164, and 187-220. The data analyzed in this study have been acquired on May 31, 2001, and consist of observations from 14 identified classes representing the land cover types in seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta. Dataset files and description can be obtained from (Datasets, ).

The experimental results are shown in Table 7 and Figure 7. From results, one can conclude that MLELM-AE greatly improved the classification accuracies with enhanced generalization capabilities for more complicated Hyperion sensor on EO-1 sensor based datasets. Evaluating Hyperion sensor on EO-1 sensor datasets is more challenging classification problem dominated by complex urban classes and nested regions than AVIRIS and ROSIS. The detailed accuracy analysis in terms of average, overall, and kappa accuracies along with the time taken to train and test the model is provided in the caption.

Table 7: Classification accuracy ( $\kappa$ ) analysis and statistical measures for Botswana Dataset.

Class Names	(Train, Test)	$\kappa$	Recall	Precision	F1-Score
Water	(81, 270)	0.9989 $\pm$ 0.0021	1.000	1.0000	1.0000
Hippo Grass	(31, 101)	0.9757 $\pm$ 0.0187	1.000	0.9859	0.9929
Floodplain Grasses1	(76, 251)	0.9897 $\pm$ 0.0057	0.9943	0.9943	0.994285714
Floodplain Grasses1	(65, 215)	0.9953 $\pm$ 0.0039	1.0000	0.9494	0.9740
Reeds1	(81, 269)	0.8787 $\pm$ 0.0132	0.8829	0.8737	0.8783
Riparian	(81, 269)	0.7011 $\pm$ 0.0204	0.7128	0.8323	0.7679
Firescar 2	(78, 259)	0.9912 $\pm$ 0.0040	0.9945	1.0000	0.9972
Island Interior	(61, 203)	0.9796 $\pm$ 0.0067	0.9859	0.9929	0.9894
Acacia Woodlands	(95, 314)	0.9347 $\pm$ 0.0135	0.9361	0.9031	0.9193
Acacia Shrublands	(75, 248)	0.9006 $\pm$ 0.0172	0.8671	0.9554	0.9091
Acacia Grasslands	(92, 305)	0.9577 $\pm$ 0.0124	0.9718	0.9039	0.9367
Short Mopane	(55, 181)	0.9190 $\pm$ 0.0187	0.9524	0.9302	0.9412
Mixed Mopane	(81, 268)	0.9337 $\pm$ 0.0145	0.9412	0.9072	0.9239
Exposed Soils	(29, 95)	0.9682 $\pm$ 0.0129	1.0000	1.0000	1.0000

In this section, we performed a set of experiments to evaluate MLELM-AE using both ROSIS, AVIRIS, and NASA EO-1 Satellite Hyperion EO-1 sensors datasets. Evaluating ROSIS and Hyperion sensors datasets are more challenging classification problems dominated by complex urban classes and nested regions than AVIRIS. Figures 1-7 and Tables 1-7 shows the overall, average, and kappa ( $\kappa$ ) accuracies along with the training and test time taken as a function of the number of labeled samples. The Figures 1-7 and Tables 1-7 are generated based on only selected samples in contrast to the entire population which reveal clear advantages of using fewer labeled samples for MLELM-AE pipeline.

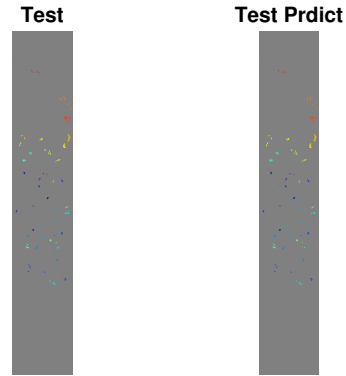


Figure 7: True Testing Maps and predicted Test Maps for Botswana datasets with 10-fold-cross-validation-based  $\kappa = 0.9268 \pm 0.0039$ , Average Accuracy =  $0.9374 \pm 0.0041$ , Overall Accuracy =  $0.9325 \pm 0.0036$ , Mean-squared Error =  $0.8309 \pm 0.0413$ , and Training Time =  $2.5815 \pm 0.0895$  and Test Time =  $0.1872 \pm 0.0182$ .

## 5 CONCLUSIONS AND FUTURE WORK

In this work we implemented a framework for hyperspectral image classification in computational efficient fashion using extreme learning machine-based autoencoder (MLELM-AE). MLELM-AE is a special case of traditional ELM where the input is equal to output and randomly generated weights are chosen to be orthogonal. The internal representation of MLELM-AE provides an effective solution not only for feed forward neural networks but also for multi-layered feed forward neural networks. MLELM-AE network provides better generalization performance than traditional back propagation based deep neural networks.

To further improve generalization in future work we will focus on learning the dictionary of each class in both the spectral and spatial domain. We will further look into the possible ways to decrease the computational complexity of the model by employing the resorting of spatial filtering (Hao et al., 2017) and extended multi-attribute profiles-based (Mura et al., 2010) methods.

## ACKNOWLEDGEMENT

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

## REFERENCES

- Ahmad, M., Alqarni, M. A., Khan, A. M., Hussain, R., Mazzara, M., and Distefano, S. (2019). Segmented and non-segmented stacked denoising autoencoder for hyperspectral band reduction. *Optik - International Journal for Light and Electron Optics*, 180:370–378.
- Ahmad, M., Bashir, A. K., and Khan, A. M. (2017a). Metric similarity regularizer to enhance pixel similarity performance for hyperspectral unmixing. *Optik - International Journal for Light and Electron Optics*, 140(C):86–95.
- Ahmad, M., haq, I. U., and Qaisaro, M. (2011). Aik method for band clustering using statistics of correlation and dispersion matrix. In *2011 International Conference on Information Communication and Management*, pages 114–1180.
- Ahmad, M., Khan, A. M., and Hussain, R. (2017b). Graph-based spatial-spectral feature learning for hyperspectral image classification. *IET Image Processing*, 11(12):1310–1316.
- Ahmad, M., Khan, A. M., Hussain, R., Protasov, S., Chow, F., and Khattak, A. M. (2016). Unsupervised geometrical feature learning from hyperspectral data. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6.
- Ahmad, M., Protasov, S., Khan, A. M., Hussain, R., Khattak, A. M., and Khan, W. A. (2018). Fuzziness-based active learning framework to enhance hyperspectral image classification performance for discriminative and generative classifiers. *PLoS ONE*, 13:e0188996.
- Arguello, F. and Heras, H. B. (2015). Elm-based spectral-spatial classification of hyperspectral images using extended morphological profiles and composite feature mappings. *Int. J. Remote Sens.*, 36(2):645–664.
- Chen, C., Li, W., Su, H., and Liu, K. (2014). Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine. *Remote Sensing*, 6(6):5795–5814.
- Datasets, H. accessed on may, 2018. [http://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes).
- Ding, S., Zhao, H., Zhang, Y., Xu, Z., and Nie, R. (2015). Extreme learning machine: Algorithm, theory and applications. *Artif. Intell. Rev.*, 44(1):103–115.
- Dora, B. H., Arguello, F., and Pablo, Q.-B. (2014). Exploring elm-based spatial spectral classification of hyperspectral images. *International Journal of Remote Sensing*, 35(2):401–423.
- Hao, L., Chang, L., Cong, Z., Zhe, L., and Chengyin, L. (2017). Hyperspectral image classification with spatial filtering and  $2,1$  norm. In *Sensors*.
- He, L., Li, J., Liu, C., and Li, S. (2018). Recent advances on spectralspatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1579–1597.
- Huang, G. B., Chen, L., and Siew, C.-K. (2006). Universal approximation using incremental constructive feed-forward networks with random hidden nodes. *Trans. Neur. Netw.*, 17(4):879–892.
- Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63.
- Johnson, W. and Lindenstrauss, J. (1984). Extensions of lipschitz maps into a hilbert space. 26:189–206.
- Kasun, L., Zhou, H., Huang, G. B., and Vong, C.-M. (2013). Representational learning with elms for big data. 28:31–34.
- Li, J., Bioucas-Dias, J. M., and Plaza, A. (2013). Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):844–856.
- Liu, C., He, L., Li, Z., and Li, J. (2018). Feature-driven active learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):341–354.
- Mura, M. D., Benediktsson, J. A., Waske, B., and Bruzzone, L. (2010). Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762.
- Ren, J., Zabalza, J., Marshall, S., and Zheng, J. (2014). Effective feature extraction and data reduction in remote sensing using hyperspectral imaging [applications corner]. *IEEE Signal Processing Magazine*, 31(4):149–154.
- Shen, Y., Xu, J., Li, H., and Xiao, L. (2016). Elm-based spectral-spatial classification of hyperspectral images using bilateral filtering information on spectral band-subsets. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 497–500.
- Zhou, Y., Peng, J., and Chen, C. L. P. (2015). Extreme learning machine with composite kernels for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2351–2360.