e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:09/September-2022 Impact Factor- 6.752 www.irjmets.com

PATCH-WISE HYPERSPECTRAL IMAGE CLASSIFICATION USING COMPOSITE 3D-2D CONVOLUTIONAL NEURAL NETWORK FEATURE HIERARCHY

Syeda Sara Samreen*1, Hakeem Aejaz Aslam*2

*1Masters Student, Department Of ECE, Muffakham JAH College Of Engineering And Technology, Hyderabad, India.

*2Assistant Professor, Department Of ECE, Muffakham JAH College Of Engineering And Technology, Hyderabad, India.

DOI : https://www.doi.org/10.56726/IRJMETS30187

ABSTRACT

Recent advances in hyperspectral imaging have increased the use of convolutional neural networks for classification. However, using only a 2D or 3D convolutional neural network necessarily involves high computational complexity and does not effectively exploit spectral spatial features. Traditional classifiers, such as Support vector machine (SVM) provide adequate classification accuracy; however, SVM cannot properly classify data when classes overlap. A large number of training samples are also required. To address the shortcomings of such traditional classifiers, a composite 3D-2D convolutional neural network feature hierarchy is developed. The proposed method exploits both the spectral and spatial features of the images very well, while the 2D convolutional layer aids in the spatial feature extraction. The dataset is divided into 3D cubes or patches, and the performance of the patches is compared using sigmoid and SoftMax loss functions, along with adaptive motion estimation (ADAM) and stochastic gradient decent (SGD) optimization. When compared to traditional classifiers such as SVM, the proposed method achieves high classification accuracy. The effect of patch sizes and training sample sizes on classification accuracy is studied. The purpose of this study is to help and encourage other researchers to conduct more research in this topic.

Keywords: Convolutional Neural Network, Hyperspectral Image Classification, Softmax, Stochastic Gradient Descent, SVM.

I. INTRODUCTION

Only recently have people been able to easily obtain high spatial resolution and high spectral resolution hyperspectral remote sensing photographs. Due to their high resolving capabilities for fine spectra, hyperspectral images have a wide range of applications in the environmental, military, mining, and medical areas [1]. Image correction, noise reduction, transformation, dimensionality reduction, and classification are the most popular techniques used to process hyperspectral remote sensing images. Hyperspectral images, unlike conventional photos, contain a lot of spectral data and this spectrum data can reveal the physical features of the object of interest, which is useful for image classification.

Hyperspectral image categorization is the most active topic of hyperspectral research. Computer classification of remote sensing images analyses and classifies the information of the earth's surface and its environment on remote sensing photographs in order to locate the features that match to the image information and extract the necessary feature information. The computer classification of remote sensing images is a specific use of automatic pattern recognition technology in the field of remote sensing. However, classification of hyperspectral pictures frequently necessitates a large amount of memory and processing effort.

So, the main motivation behind this work is to overcome drawbacks of traditional classifiers and to implement the proposed method using optimization and loss functions to improve overall classification accuracy. Convolutional Neural Network (CNN) is the most well-known and widely used deep learning method. The fundamental advantage of convolutional neural network over its predecessors is that it does it automatically, without human interference. CNNs have also been used extensively in many other domains, including machine vision, voice processing, face recognition, and so on. CNNs, like traditional neural networks, were impacted by neurons occurring in human and animal brains. Composite CNN is an amalgamation of the advantages of both 3D and 2D CNN put together in layers. We use optimization to further boost the classification performance.

The dataset is divided into patches of different sizes and into 3D cubes because it can facilitate better exploitation of both the spectral and spatial features of the hyperspectral images. Indian Pines dataset is used for the research. The data is divided into 10%, 20%, 30% training data and the effect of different window or patch sizes on accuracy and training time is also investigated.

II. RELATED WORK

Conventional hyperspectral Image classification methods mostly rely on spectral data. These traditional methods involve steps to obtain the features from input data and secondly to apply a classifier to learn the features. These mainly include classifiers based on like k- nearest neighbours [2], logistic regression [3], Support Vector Machine [4] and maximum likelihood [5]. These classification algorithms do not capture the spatial variation in the hyperspectral images and also face dimensionality issues. Dimensionality reduction techniques are suggested to overcome these issues and for effective processing of data. It is however difficult to determine the best dimensionality reduction technique. Some of the dimensionality reduction techniques like independent component analysis (ICA) [6], linear discriminant analysis [7] and principal component analysis [8,9] are generally used as initial steps to extract better feature with less dimensions or components.

In recent times deep learning also proved to be efficient in hyperspectral imaging and its classification. Deep learning allows automatic learning of features and is robust [10]. There are many deep learning techniques like recurrent neural networks (RNN) [11], deep belief network (DBN) [12], contractive auto encoder (CAE) [13] that provide better classification performance and higher learning rates [14].

CNNs have recently surpassed other standard approaches in deep learning application on GPUs [15]. CNNs, a newer method allowing hyperspectral image classification, are typically used for visual-related problems. In a convolutional neural network, linear convolution filters are followed by nonlinear activation functions to generate spectral and spatial feature maps.

The author of [16] proposes a large network with high restrictions to handle the HSI and classification task with few training data. The huge feedforward DNN utilizing deep three-dimensional CNN with simulated inputs delivers by far the best results. In recent years, there has been a significant advance Hyperspectral image classification [17], where the spatial characteristics are adjusted by a 2D CNN architecture [18], [19]. However, these spatial elements are often retrieved individually, which negates the need to simultaneously use spatial-spectral features for classification of hyperspectral images. To overcome these drawbacks the author in [20] proposed a 3D-CNN HSI classification framework that makes full use from both spectral and spatial data included within HSI to improve HSI classification and involves fewer parameters compared to other method of hyperspectral image classification. It is however evident from the above literature that using only two dimensional or three dimensional has certain drawbacks, such as lost channel link information or a very complicated model.

It also hindered these approaches from reaching higher HSI accuracy. The major reason for this is that HSIs are volumetric data with a spectral dimension. The 2-D-CNN cannot generate good distinguishing feature maps on its own using the spectral dimensions. Likewise, a deep 3-D-CNN is much more computationally complicated, which appears to degrade performance for classes with comparable textures over several spectral bands.

In this paper, we aim to overcome the short comings of the previous methods and combine the advantages of both 2D and 3D CNN into composite model to exploit both spectral and spatial feature to their maximum. Iterative training and optimizers like stochastic gradient descent and are implemented to reduce training loss. Loss functions like Sigmoid and SoftMax are utilized to quantify how well is the classification performance of the model. The composite CNN model is applied to Indian Pines dataset and the accuracy for different patches or window sizes of input data is observed. The model is trained for different training sample sizes and the accuracy is compared for all the cases.

III. PROPOSED METHOD

Let $Q \in R^{WXHXB}$ represent the hyperspectral cube. Every pixel in Q comprises B spectral measurements. The mixed land-cover classes in the hyperspectral pixels introduce considerable intraclass variability and interclass similarity into Q. Taking on this topic is a massive undertaking for any model. To address redundancy, we use PCA to reduce the number of spectral bands from B to N while keeping the dimensions WXH constant. We simply decreased spectral bands to maintain spatial information, which is crucial for object recognition. The

PCA reduced data cube is represented by Y∈ R^{WXHXN}, where Y represents the modified input after dimensionality reduction, W represents the width, H represents the height, and N is the number of spectral bands after PCA.

The Hyperspectral image data cube is then separated into 3D patches that overlap each other in order to perform image categorization algorithms. A 3D adjacent patch, X ∈ P^{SXSXN}, is generated from Y and is centered at a spatial point (u, v) that covers SXS windows. The overall number of patches produced by Y is provided by (W-S+1) X (H-S+1).The 3D patch at (u, v) is depicted as $X_{u, v}$ with widths ranging between U-(S-1)/2 to U+(S-1)/2 and heights varying from V-(S-1)/2 to V+(S-1)/2, as well as all N spectral bands produced post applying PCA.

In 2D CNN, the input data is convolved with 2-D kernels. Convolution happens by adding the dot product of the input information with the kernel. The kernel is shifted across the input data to span the whole spatial dimension. Nonlinearity is introduced into the model by feeding the convolved features via the activation function. The jth feature map in the ith layer's activation value at a spatial point (c, d) is given by $\nu_{i,j}^{c,d}$ obtained from Equation 1.

$$\nu_{i,j}^{c,d} = \theta(b_{ij} + \sum_{\tau=1}^{d_{l-1}} \sum_{\rho=-\alpha}^{\alpha} \sum_{\beta=-\delta}^{\delta} \omega_{i,j,\tau}^{\beta,\rho} \times \nu_{i-1,\tau}^{c+\beta,d+\rho}) \ldots\ldots\ldots\ldots \quad (1)$$

Where θ is the activation function, ω_{ij} is the kernel depth for the jth feature map at ith layer, d_{l-1} is the number of feature maps in (l-1)th layer, b_{ij} is the bias parameter for the jth feature map of the ith layer, and $2\alpha + 1$ signifies the width and $2\beta + 1$ height of the kernel, respectively.

Similarly, in 3D convolution, the activation value at the spatial position (c, d, e) in the jth feature map of the ith layer is represented by $\nu_{i,j}^{c,d,e}$ as shown in Equation 2

$$\nu_{i,j}^{c,d,e} = \theta(b_{ij} + \sum_{\tau=1}^{d_{l-1}} \sum_{\lambda=-\eta}^{\eta} \sum_{\rho=-\alpha}^{\alpha} \sum_{\beta=-\delta}^{\delta} \omega_{i,j,\tau}^{\beta,\rho,\lambda} \times \nu_{i-1,\tau}^{c+\beta,d+\rho,e+,\lambda}) \ldots\ldots\ldots\ldots (2)$$

Here, $2\eta + 1$ is the depth of kernel along a spectral dimension and all the other parameters are same as equation above.

Optimizers help determine how to alter both the learning rate and the weights of the model to minimize losses. The parameters are updated at each training sample while optimizing with Stochastic Gradient Descent (SGD) or Adaptive moment estimation (ADAM). Before training, it is recommended to freely sample the training data in each epoch. This method is more memory-efficient for big training datasets.

ADAM optimizer’s method is to compute an adaptive learning rate for each component or parameter of the model. It is computationally efficient and straightforward to implement.

The output layer, which is the outermost layer of the CNN design, completes the final process. Certain loss functions are employed in the CNN model for the output layer to calculate the predicted error created over the training data. Sigmoid and SoftMax loss functions are applied in this research.

Sigmoid functions are limited, differentiable, and real functions with non-negative derivatives through all real input values. This is a logistic function having a range of output values from 0 to 1. A multidimensional variation of the sigmoid function is the SoftMax loss function. The mathematical function that converts a numerical vector to a probability vector. It normalizes the outputs for each class between 0 and 1, then divides by their sum to calculate the chance that the input value corresponds to a specific class.

Stochastic Gradient Descent and Adam optimizer

Optimizers are techniques or approaches that are used to lower an error function (loss function) or to enhance production efficiency. Optimization techniques are mathematical functions that are affected by the model's learnable parameters, such as Weights and Biases. Optimizers assist in determining how to modify both learning rate as well as weights of a neural network to minimize losses.

In SGD, the parameters are modified at each training sample. It is preferable to randomly sample the training data in each epoch before to training. This approach is both more memory-efficient for large-sized training datasets. Unfortunately, because it is often updated, it makes incredibly noisy steps toward the solution, causing the convergence pattern to become exceedingly unstable. SGD with Momentum is stochastic based optimization approach that augments ordinary stochastic gradient descent with a momentum factor. Momentum replicates an object's inertia while moving; that is, the previous update direction is kept to some level during the update, whilst present update gradient is utilized to fine-tune the ultimate update direction. In this method, you may boost steadiness to a certain level, allowing you to learn faster while also eliminating the need for local optimization. In this project momentum of 0.9 is used. It helps to reduce noise however; the drawback is that it adds an extra hyperparameter.

ADAM is another popular optimization approach or learning process. ADAM exemplifies the most recent deep learning optimization trends. The Hessian matrix, with a second-order derivative, represents this. Adam is a learning technique created specifically for training deep neural networks. Adam has two advantages: more memory efficiency and less processing power. Adam's approach is to compute an adaptive learning rate corresponding to every component or parameter of the model. It is computationally efficient and easy to implement. Adam's equation is represented by equations (3- 4) as shown.

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{S_{d\omega_t} - \varepsilon}} * V\, dw_t \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$b_t = b_{t-1} - \frac{\eta}{\sqrt{S_{db_t} - \varepsilon}} * V\, db_t \ \ldots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Sigmoid and softmax loss function

The last step is accomplished by the output layer, which is the final layer of the CNN design. In the CNN model, certain loss functions are used for the output layer to determine the expected error generated over the training samples. This mistake displays the discrepancy between the actual and projected output. The CNN learning procedure will then be used to optimize it.

Sigmoid functions are restricted, differentiable and real functions that have a non-negative derivative through each point and are specified for all realistic input values. This is a function that is a logistic function with an output range from 0 to 1. When compared to the linear function (inf, inf), the output of this function will always be in the range (0,1). It has a fixed output range, is non-linear, along with being continuously differentiable, and monotonic. However, it is not zero-centered and is represented by equation 5

$$S(x) = \frac{1}{1+e^{-x}} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The limitations of sigmoid activation function are the vanishing gradient problem, it is not zero centric function and is computationally expensive. The SoftMax loss function is a multidimensional version of the sigmoid function. It is the mathematical function that turns a numerical vector into a probability vector. When dealing with multi-class classification challenges in machine learning, SoftMax is often used as an activation function. Its output is regarded as the likelihood of obtaining each class. Advantages of SoftMax function are that it is suitable for multiclass categorization. It normalizes the outputs between 0 and 1 for every class, then carries out division by their sum to determine the likelihood that perhaps the input value belongs to a given class. Also, SoftMax, with reference to equation 6 is frequently used only for the output nodes in neural networks that require to classify inputs into many categories.

$$\text{SoftMax (Zi)} = \frac{\exp \text{Zi}}{\sum \exp(\text{Zi})} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

The confusion matrices are utilized to construct the Average Accuracy (AA), Overall Accuracy (OA), and Kappa coefficients for assessment purposes. AA is the average of classification performance (class wise), OA is the number of successfully classified instances out of the total test examples, and is a statistical metric that takes into account mutual information about a good agreement between classification map versus ground-truth maps. Along with OA, AA, and metrics, numerous statistical tests such as F1-Score, Precision, and Recall are being evaluated. The proposed composite 3D-2D CNN model represented in fig.1 has parameters, commonly known as adjustable weights. The weights are first randomized before being modified using a stochastic gradient descent (SGD) optimizer and sigmoid loss function. The weights are modified using a 256-epoch mini-batch. SGD and ADAM optimizer with sigmoid loss function and SoftMax loss function is used to perform comparative research. The dataset is separated into different training and testing sample sizes, and the influence of different training sample sizes is investigated. The process flowchart of the work is represented in fig.2.

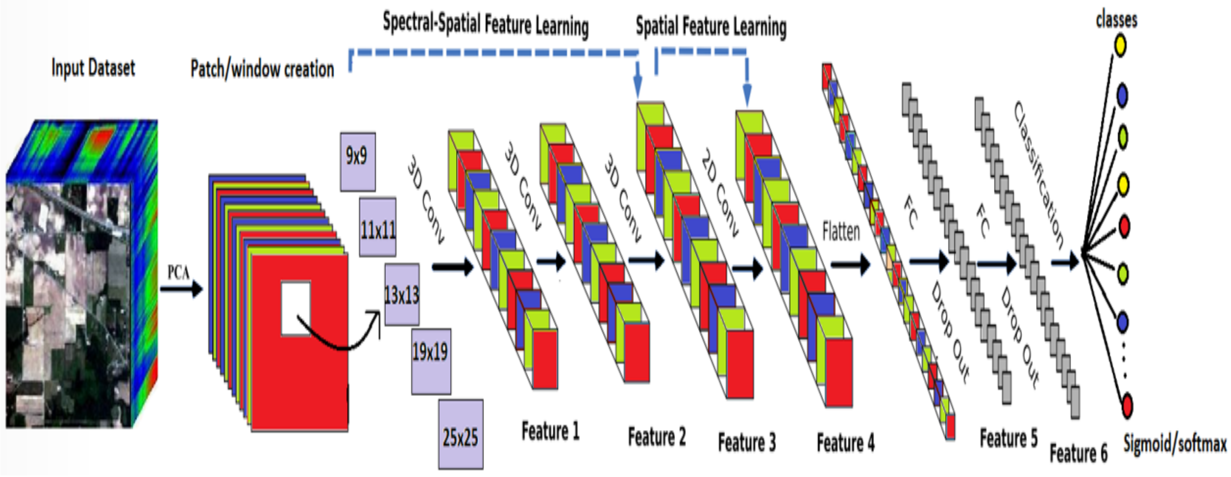


Figure 1: Architecture of proposed method

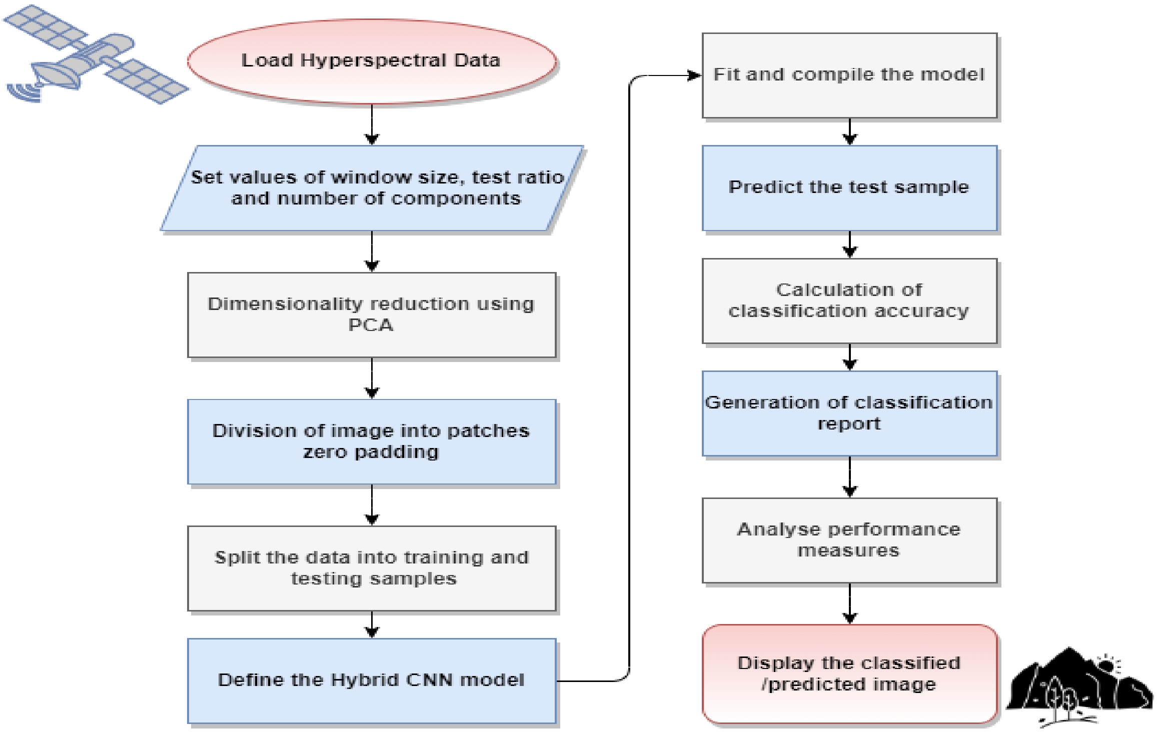

Load Hyperspectral Data
Set values of window size, test ratio and number of components
Dimensionality reduction using PCA
Division of image into patches zero padding
Split the data into training and testing samples
Define the Hybrid CNN model
Fit and compile the model
Predict the test sample
Calculation of classification accuracy
Generation of classification report
Analyse performance measures
Display the classified /predicted image

Figure 2: Block Diagram for proposed method

IV. DATASET

The Indian Pines (IP) dataset was taken over northern Indiana's test facility, Indian Pines, by an AVIRIS sensor and consists of 145x145 pixels and 224 bands with wavelengths ranging from 0.4- 2.5 x 10^{-6} meters. The number of spectral bands is reduced to 224 by eliminating the water absorption bands. Two-thirds of this dataset is agricultural, while one-third is forest and other naturally evergreen vegetation. This dataset also includes a railway line, two dual-lane motorways, low-density buildings, residences, and tiny roads. Furthermore, certain crops in their early phases of development are present, accounting for less than 5% of overall coverage. It's ground truth consists made up of 16 classes, but they are not completely exclusive. Table 1 represents the number of classes and the corresponding labels for the dataset considered. Fig. 3 represents the ground truth of Indian Pines (IP) dataset.

Table 1: Classes and samples of IP Ground truth

Class no	Label	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pastured-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soyabean-notill	972
11	Soyabean-mintill	2455
12	Soyabean-clean	593
13	Wheat	205
14	Woods	1265
15	Building-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93

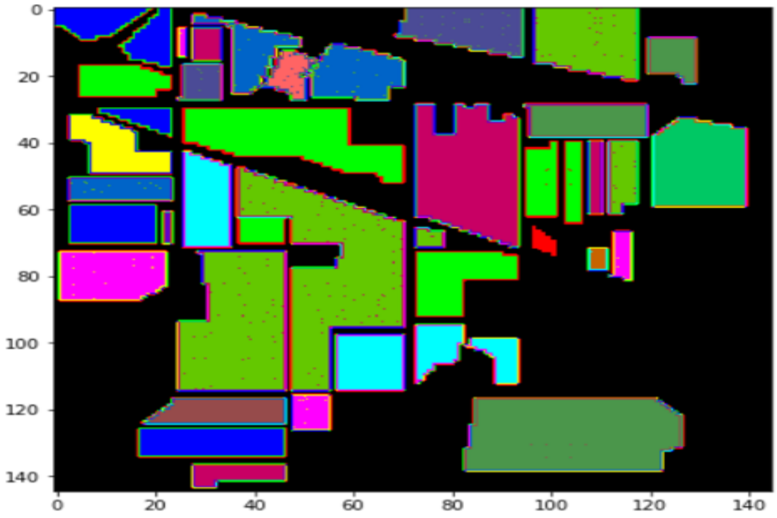


Figure 3: IP Ground Truth

Hyperparameters

Table 2 represents the hyperparameter used to obtain the results for this paper. It includes details of which optimization algorithm is used, the loss functions, the learning rate, the different patch sizes used and others.

Table 2: Hyperparameters used in experiments

Hyperparameter	Value
Dataset	Indian Pines Dataset
Window size	9x9,11x11, 13x13, 19x19, 25x25
Test Ratio	0.7,0.8,0.9
Spectral bands	30

Learning rate	0.001
Momentum	0.9
Decay	1e-06
Dropout	0.4
Optimization Algorithm	SGD, Adam
Testing Loss Function	Sigmoid, SoftMax
Training mini batch size	256
Number of epochs	80

V. RESULTS AND DISCUSSION

Experiment 1: Effect of spatial window sizes on classification accuracy

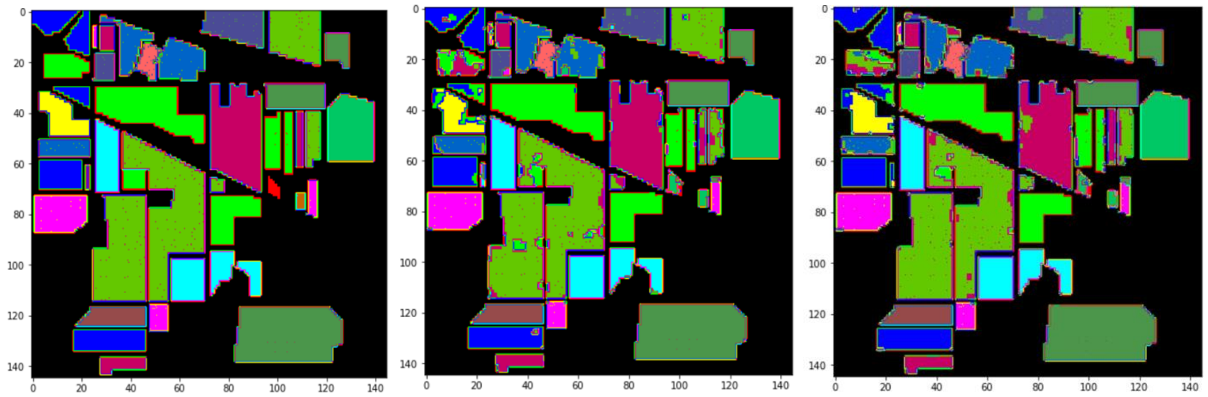
Using stochastic gradient descent optimizer and SoftMax loss function:

The impact of different spatial window size is examined by using different patch sizes input for classification. The different window sizes used are 9x9, 11x11, 13x13, 19x19, 25x25. If the window size is set too small, it reduces inter-class variation in samples, and if the window size is set too big, it may take in pixels from many classes, resulting in misclassification in both circumstances. The classification performance is obtained using the proposed method for 80 epochs using stochastic gradient descent optimizer and SoftMax loss function. It is observed that with increase in patch size the overall accuracy increases and the test loss decreases.

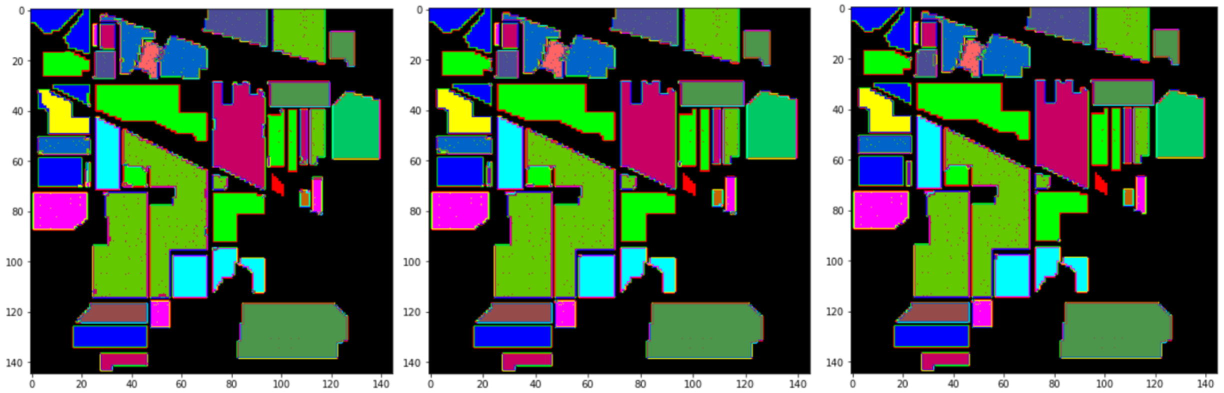
Table 3 represents classification performance for different patch sizes using SGD optimization and SoftMax loss function. Fig.4 represents the predicted classification map for different window sizes using stochastic gradient descent optimizer and SoftMax loss function.

Table 3: Classification performance for different patch sizes using SGD optimization and SoftMax loss function.

Parameters	**9x9**	**11x11**	**13x13**	**19x19**	**25x25**
Total Accuracy	88.571	95.219	97.477	99.512	99.679
Total loss	35.392	14.508	10.811	1.853	1.532
Overall Accuracy	88.571	95.219	97.477	99.512	99.679
Kappa	86.908	94.548	97.120	99.443	99.634
Average Accuracy	73.102	87.716	90.675	98.763	99.749



(a) (b) (c)



(d) (e) (f)

Figure 4: Predicted classification maps for different patch sizes using stochastic gradient descent optimizer and SoftMax loss function. (a) Ground Truth of Indian Pines (b) 9x9 (c) 11x11 (d) 13x13 (e) 19x19 (f) 25x25.

Using Stochastic gradient descent optimizer and Sigmoid loss function:

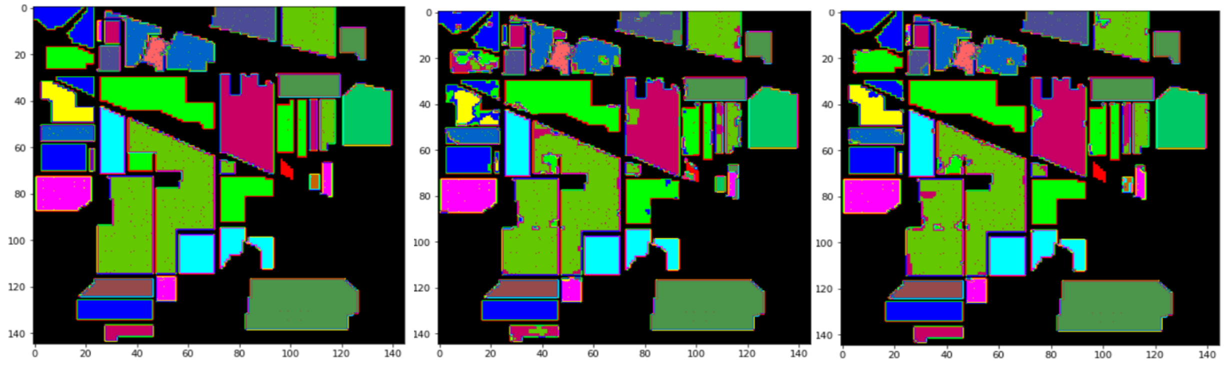
The classification performance is obtained using the proposed method for 80 epochs using stochastic gradient descent optimizer and sigmoid loss function. Sigmoid loss function is observed to perform better in terms of computational time. The computational time is reduced and the overall classification accuracy is improved. Also, it is observed that 19x19 window size in this case is optimum to achieve better accuracy.

Table 4 represents the classification performance of the Indian pines dataset in terms of overall accuracy, kappa and average accuracy for this case.

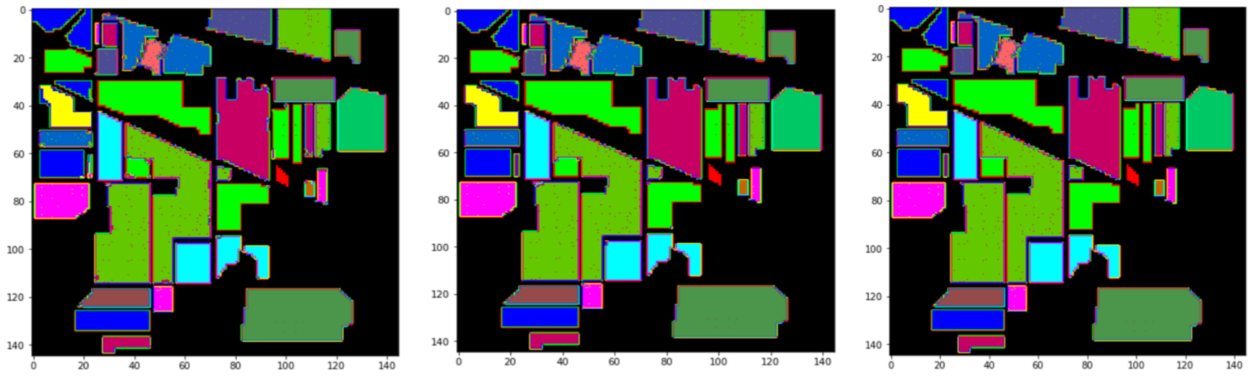
Fig.5 represents the predicted classification map for different window sizes using stochastic gradient optimizer and sigmoid loss function.

Table 4: Classification performance for different patch sizes using SGD optimization and Sigmoid loss function.

Parameters	9x9	11x11	13x13	19x19	25x25
Total Accuracy	90.717	96.278	98.425	99.456	99.790
Total loss	27.570	12.367	5.553	1.770	0.910
Overall Accuracy	90.717	96.222	98.425	99.456	99.435
Kappa	89.383	95.695	98.204	99.380	97.084
Average Accuracy	77.455	89.734	93.191	97.999	98.960



(a) (b) (c)



(d) (e) (f)

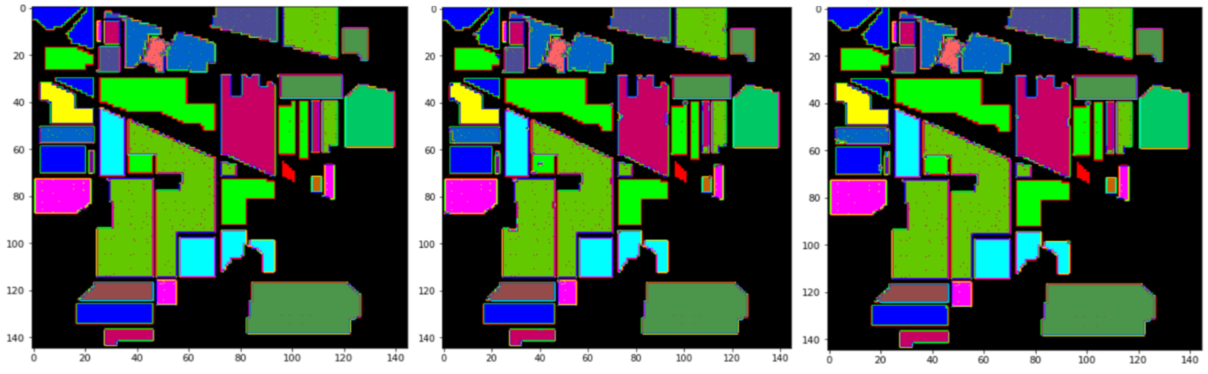
Figure 5: Predicted classification maps for different patch sizes using stochastic gradient descent optimizer and Sigmoid loss function. (a) Ground Truth of Indian Pines (b) 9x9 (c) 11x11 (d) 13x13 (e) 19x19 (f) 25x25.

Using ADAM optimizer and SoftMax loss function:

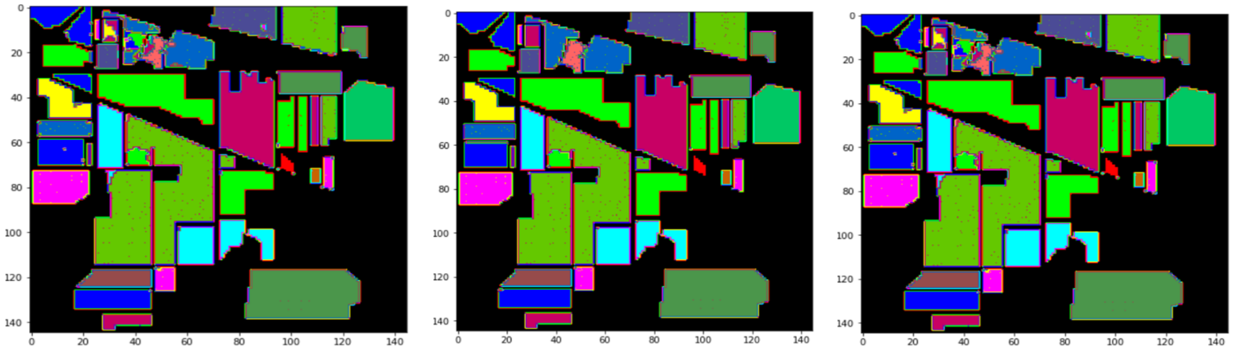
The classification performance is obtained using the proposed method for 80 epochs using ADAM optimizer and SoftMax loss function in order to further push the classification accuracy and reduce the time taken for computation. ADAM optimizer requires less parameters for tuning and is faster computationally. Table 5 represents the classification performance of the Indian pines dataset in terms of overall accuracy, kappa and average accuracy for this case. Fig.6 represents the classification results for different patch sizes using ADAM optimizer and SoftMax loss function.

Table 5: Classification performance for different patch sizes using ADAM optimization and SoftMax loss function.

Parameters	9x9	11x11	13x13	19x19	25x25
Total Accuracy	98.648	99.386	99.526	99.540	99.756
Total loss	7.136	4.263	2.662	2.149	4.320
Overall Accuracy	98.648	99.386	99.526	99.540	99.760
Kappa	98.458	99.300	99.459	99.475	98.855
Average Accuracy	97.322	98.613	99.181	99.718	99.072



(a) (b) (c)



(d) (e) (f)

Figure 6: Predicted classification maps for different patch sizes using ADAM optimizer and SoftMax loss function. (a) Ground Truth of Indian Pines (b) 9x9 (c) 11x11 (d) 13x13 (e) 19x19 (f) 25x25.

Fig. 7 depicts the findings, where (x-y) reflects the classification accuracy gained by feeding the model different patch sizes while using different optimizers and loss functions as mentioned in experiment 1, with x and y signifying patch sizes and overall accuracy, respectively.

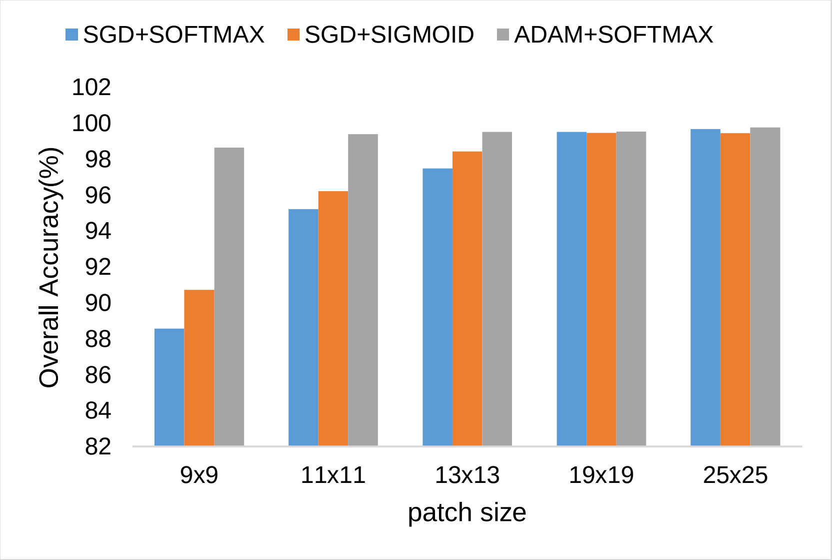


Figure 7: Influence of patch sizes on classification accuracy

The overall accuracy is observed to increase with increase in patch size. However, it seems to plateau at patch size 25x25.The model is observed to perform consistently well while using ADAM optimizer and SoftMax loss function.

Experiment 2: Effect of spatial window sizes on training time

The impact of different spatial window size on training time is examined by using different patch sizes input for classification.

The different window sizes used are 9x9, 11x11, 13x13, 19x19, 25x25.It is observed that with increase in patch size the time taken to train the proposed model increases. Table 6 represents the training time in seconds obtained from the three considered scenarios in the above experiment. Fig.8 represents the training time required for the three cases considered in experiment 1 where x represents the patch sizes and y represents the training time in seconds. It is observed that as the patch size increases the time taken to train the model also increases. The training time required to train the model using SGD optimizer and sigmoid and SoftMax loss function is almost equivalent when the patch size increases from 13x13 to 25x25.

Table 6: Training time for different window sizes in different cases

Methods	9x9	11x11	13x13	19x19	25x25
SGD + SoftMax	423.4	1014.1	1887.2	4359.7	10379.3
SGD+ Sigmoid	352.7	733.4	1276.7	4965.3	10694.8
ADAM+ SoftMax	322.3	721.2	1244.8	6233.5	14329.6

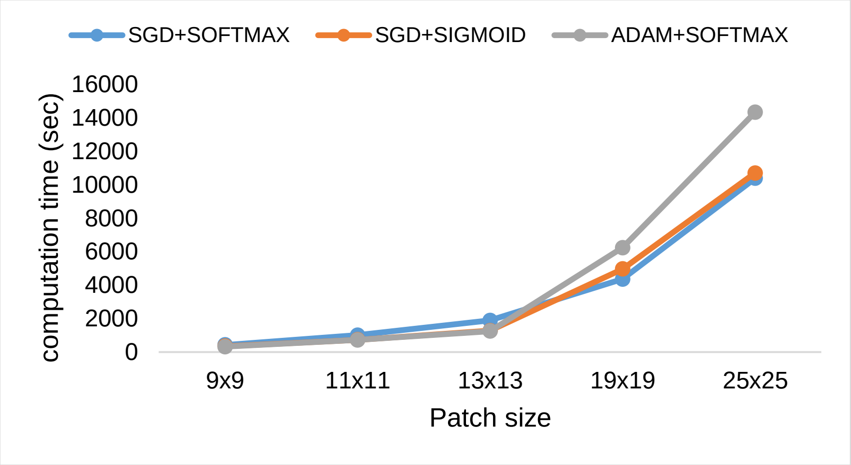


Figure 8: Influence of patch sizes on training time

Experiment 3: Effect of training sample size on classification accuracy

This experiment investigates the influence of altering training testing sample sizes on classification accuracy. The input is given in the form of patches of varying sizes. The window sizes utilised are 9x9, 11x11, 13x13, 19x19, and 25x25. In this scenario, ADAM optimization is applied, and SoftMax loss function is used in the suggested model's final layer. In Fig.9, the overall accuracy is compared for three scenarios where different percentages of training samples are considered i.e., 30%,20%,10%. It is observed that the larger the training sample size the better is the classification accuracy. With decrease in training sample size the overall classification accuracy decreases too. Table 7 represents the overall accuracy obtained for different training sample sizes.Fig.10 depicts the classification report for 25x25 patch size with30% training sample size.

Table 7: Overall accuracy for patches using different training sample sizes.

Training sample size	9x9	11x11	13x13	19x19	25x25
30%	98.64	99.38	99.52	99.54	99.89
20%	97.18	98.51	99.23	98.85	99.15
10%	94.78	96.29	97.97	98.31	98.20

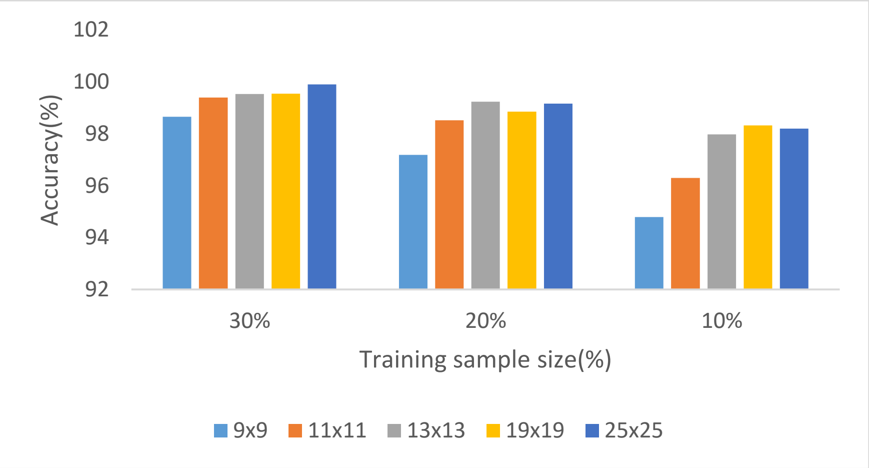
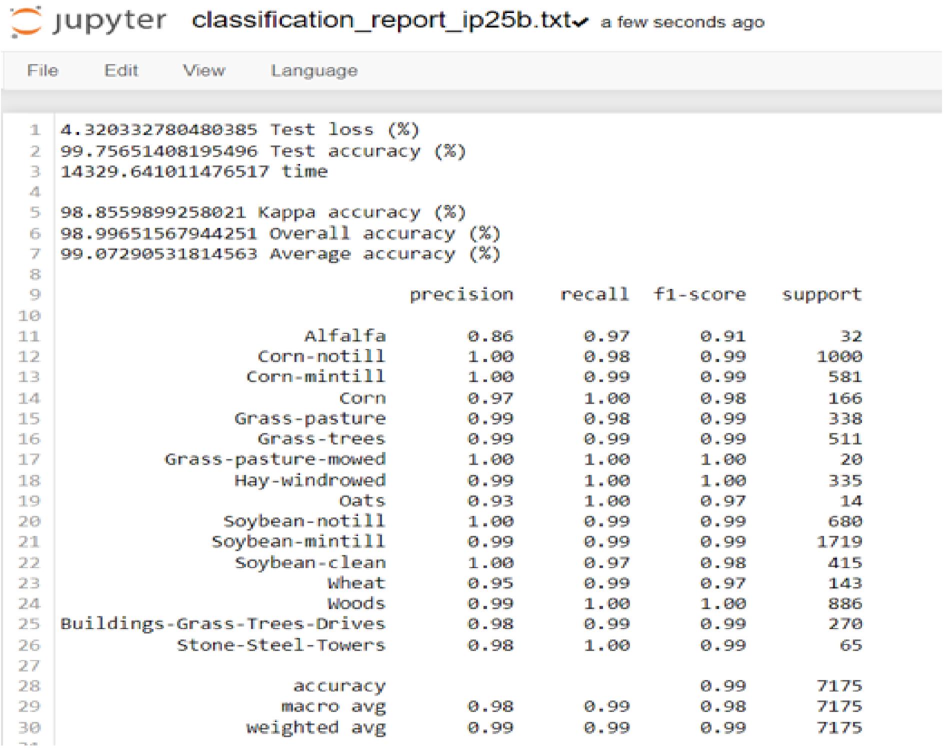


Figure 9: Influence of different training sample sizes on classification accuracy.



```
4.320332780480385 Test loss (%)
99.75651408195496 Test accuracy (%)
14329.641011476517 time

98.8559899258021 Kappa accuracy (%)
98.99651567944251 Overall accuracy (%)
99.07290531814563 Average accuracy (%)

                              precision    recall  f1-score   support

                     Alfalfa       0.86      0.97      0.91        32
                 Corn-notill       1.00      0.98      0.99      1000
                Corn-mintill       1.00      0.99      0.99       581
                        Corn       0.97      1.00      0.98       166
               Grass-pasture       0.99      0.98      0.99       338
                 Grass-trees       0.99      0.99      0.99       511
         Grass-pasture-mowed       1.00      1.00      1.00        20
               Hay-windrowed       0.99      1.00      1.00       335
                        Oats       0.93      1.00      0.97        14
              Soybean-notill       1.00      0.99      0.99       680
             Soybean-mintill       0.99      0.99      0.99      1719
               Soybean-clean       1.00      0.97      0.98       415
                       Wheat       0.95      0.99      0.97       143
                       Woods       0.99      1.00      1.00       886
Buildings-Grass-Trees-Drives       0.98      0.99      0.99       270
          Stone-Steel-Towers       0.98      1.00      0.99        65

                    accuracy                           0.99      7175
                   macro avg       0.98      0.99      0.98      7175
                weighted avg       0.99      0.99      0.99      7175
```

Figure 10: Classification report of 25x25 patch size for 30% training sample data

VI. CONCLUSION

This study proposes a composite 3D-2D CNN model that shows exceptional HSI classification performance on the Indian Pines dataset while being computationally efficient. It exhibits decent performance even with less amount of training data. The main focus of this study is to observe the three different optimizer and loss function combinations and obtain classification accuracy for each case.

The Indian Pines dataset is partitioned into several patch sizes i.e., 9x9,11x11,13x13,19x19,25x25. The influence of patch sizes on classification accuracy and training time is examined. From the experiments it is observed that, larger the window size, the better the accuracy. When comparing different loss functions applied at the final layer of the proposed model, the combination of ADAM optimizer and SoftMax loss function provides the maximum accuracy of 99.7% for 25x25 window size.

The time required to train the suggested approach is investigated for window sizes of 9x9, 11x11, 13x13, 19x19, and 25x25. It has been noticed that the larger the window size, the longer the time required for training. On examination it can be observed that even though the combination of ADAM optimizer and SoftMax loss function provides the maximum accuracy of 99.7% for 25x25 window size, the combination of SGD optimizer and SoftMax loss function trains the proposed model faster by 3950.2 seconds and achieves a classification accuracy of 99.6% for 25x25 patch size input.

The influence of training sample size on classification accuracy demonstrates that the higher the training sample size, the greater the accuracy. When the training sample size is increased from 10% to 30%, the accuracy rises by 1.6%. In conclusion, the proposed model indicates that for better classification accuracy and faster training time the combination of SGD optimizer and applying SoftMax loss function in the last layer of the model provides optimum results.

Future work might include data augmentation to provide a better training set. It can assist to reduce the cost of gathering tagged hyperspectral pictures while also increasing training speed. To enhance accuracy furthermore, the model can be trained for a longer number of epochs and more layers can be added.

VII. REFERENCES

[1] Wenjing,L.v.: Xiaofei, Wang. Overview of hyperspectral image classification. Hindawi Journal of Sensors. Vol 2020. https://doi.org/10.1155/2020/4817234c

[2] B, Tu.; J,Wang.; X ,Kang.; G, Zhang.; X. Ou. and L, Guo. KNN-Based Representation of Super pixels for

Hyperspectral Image Classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2018; Volume 11, pp. 4032-4047.

[3] Li, J.; Bioucas-Dias, J.M.; Plaza A. Semi supervised hyperspectral image segmentation using multinomial logistic regression with active learning. IEEE Trans. Geoscience. Remote Sensing. 2010, Volume 48, pp.4085–4098.

[4] Y, Wang.; W, Yu.; Z, Fang. Multiple kernel-based SVM Classification of Hyperspectral Images by Combining Spectral, Spatial, and Semantic Information. Multidisciplinary Digital Publishing Institute-Remote Sensing, Vol. 12, Jan.2020, pp.120.

[5] Ediriwickrema J, Khorram S," Hierarchical maximum-likelihood classification for improved accuracies," IEEE Trans. Geoscience Remote Sensing, 1997; Vol.35, pp.810–816.

[6] Pooja VS, Marriappan V.N," Classification of hyperspectral images using principal component and independent component analysis,"J Adv Res GeoSci Rem Sens.Vol.4, November 2017, pp.14-24.

[7] Joyoshree Ghosh, Shaon Bhatta Shuvo. Improving Classification Model's Performance Using Linear Discriminant Analysis on Linear Data in proceedings of the 10th International Conference on Computing, Communication and Networking Technologies, Kanpur, India.2019.

[8] Q Sun, X Liu, M Fu. Classification of hyperspectral image based on principal component analysis and deep earning in proceedings of the 7th International Conference on Electronics Information and Emergency Communication, Shenzhen, China. 2017, pp. 356-359.

[9] Jolliffe IT, Cadima J," Principal component analysis: a review and recent developments," Phil. Trans. R. Soc. A. 2016; 374, 2065.

[10] Zhang L, Du B. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geoscience Remote Sensing Mag. 2016; Vol.4, pp. 22–40.

[11] Mou ,L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing.2017; vol. 55, pp.3639-3655.

[12] Zhong P, Gong Z, Li, Schönlieb C.B. Learning to diversify deep belief networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing.2017; Volume.55, pp.3516-3530.

[13] Hassanzadeh A, Kaarna A, Kauranne T. Unsupervised multi-manifold classification of hyperspectral remote sensing images with contractive Autoencoder in proceedings of the Scandinavian Conference on Image Analysis, Tromso, Norway. Springer, Cham.2017; pp. 169-180.

[14] Syeda Sara Samreen, Hakeem Aejaz Aslam. Hyperspectral image classification using deep learning techniques: A Review. SSRG International Journal of Electronics and Communication Engineering.2022; vol.9, pp.1-4.

[15] Hinton G E, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science,2006, Vol. 313(5786), pp.504-507.

[16] Y Chen, H Jiang, C Li, X Jia, P Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans. Geoscience Remote Sensing.2016, Volume.54, pp. 6232–6251.

[17] B Fang, Y Bai, Y Li. Combining spectral unmixing and 3d/2d dense networks with early-exiting strategy for hyperspectral image classification. Multidisciplinary Digital Publishing Institute Remote Sensing. 2020; Vol.12, pp. 779.

[18] Y Li, L He. An improved hybrid CNN for hyperspectral image classification. International Society for Optics and Photonics. SPIE.2020; 11373, pp.485-490.

[19] L.Huang, Y.Chen. Dual-path Siamese CNN for hyperspectral image classification with limited training samples. IEEE Geoscience and Remote Sensing Letters.Volume18, Issue 3, March 2021, pp.1–5.

[20] Muhammad Ahmad, Sidrah Shabbir, Rana Aamir Raza, Manuel Mazzara, Salvatore Distefano, Adil Mehmood Khan. Hyperspectral Image Classification: Artifacts of Dimension Reduction on Hybrid CNN. Optik-International Journal for Light and Electron Optics.2021, Volume 246.

https://doi.org/10.1016/j.ijleo.2021.167757.