

Hyperspectral Image Classification using Convolutional Neural Networks

Shambulinga M¹, G. Sadashivappa²

Dept. of Electronics and Telecommunication Engineering
RV College of Engineering, Bengaluru

Abstract—Hyperspectral image is well-known for the identification of the objects on the earth's surface. Most of the classifier uses the spectral features and does not consider the spatial features to perform the classification and to recognize the various objects on the image. In this paper, the hyperspectral image is classified based on spectral and spatial features using a convolutional neural network (CNN). The hyperspectral image is divided into a small number of patches. CNN constructs the high level spectral and spatial features of each patch, and the multi-layer perceptron helps in the classification of image features into different classes. Simulation results show that CNN archives the highest classification accuracy of the hyperspectral image compared with other classifiers.

Keywords—Convolutional neural network; hyperspectral image; classification

I. INTRODUCTION

Latest developments in optics and photonics have made a hyperspectral imaging sensor with better spectral and spatial resolution. The spatial and spectral information is efficiently exploited to identify the materials and objects on the earth surface. The spectral signatures are modelled in such a way that they will differentiate the various objects and materials. It is possible to see the identification of different materials, objects and surface ground cover classes based on their reflectance properties as a classification task, i.e. the classification of image pixels based on their spectral characteristics. Classification of hyperspectral imaging has been used in a broad range of applications, such as agriculture, environmental science, astronomy, surveillance, astronomy and biomedical imaging. However, the classification of the hyperspectral images has its own particular problems, in addition to (i) high dimensionality, (ii) the small number of samples which have been labelled, and (iii) significant spatial variation of spectral signatures.

Much of the recent work on hyperspectral data classification follows the traditional pattern recognition method, which comprises of two distinct steps: first, detailed handcrafted features are obtained from the original data input, and secondly, classifiers like Support Vector Machines (SVM), Neural Networks (NN) [1], maximum likelihood [2], parallelepiped classification, k nearest neighbours [3], minimum distance [4], and logistic regression [5] have been used to learn the extracted features. The “curse of dimensionality” affects the majority of the algorithms mentioned above. Any dimensionality reduction-based classification approaches [6] have been suggested to manage

the large dimension complexity and small training samples of hyperspectral data. Band selection and transformation are the other methods available to deal with dimensionality. In general, statistical learning techniques have been used to solve the large dimensionality and variability of high dimensional hyperspectral data and when few training samples are accessible.

Support vector machine, a popular classification method for hyperspectral data classification, is presented in [7]. SVM is resistant to the Hughes phenomena and has poor sensitivity to high dimensionality. In certain situations, SVM-based classifiers perform better compared to other commonly used pattern recognition strategies in terms of classification accuracy. These classifiers were cutting edge technological tools for a long time.

In recent years, spatial information has become highly relevant for hyperspectral data classification. In terms of efficiency, spatial-spectral classification methods provide substantial benefits. Many new methods aim to bring spatial information into account to deal with spectral signature spatial variability. In [8], applying SVM and a guided image filter, a technique for classifying hyperspectral images has been presented. The guided image filter is used to incorporate the spatial features into the SVM classifier. In [9], the edge-preserving filters such as bilateral filter and guided image filter are included to incorporate the spatial features to the SVM classifier.

However, it is essential to know which features are most relevant for the classification algorithms due to the high diversity of represented materials. The various deep learning models [10,11] have been developed for classification purposes. These models are trained from various levels of features. The high levels features required for training the model are obtained from low-level features. These models automate the extraction of features for any problem compared with any convolutional pattern recognition technique. In addition, deep learning systems tend to match and resolve the classification issue more efficiently with wider datasets and large images with high spatial and spectral resolution. Deep learning approaches have already shown promising results for detecting real artefacts, such as man-made objects or vehicles, as well as for classifying hyperspectral data.

More precisely, a deep learning system for the classification of hyperspectral data with accurate results was used in [12].

In precise, the auto encoder's helps in the design of the deep architecture, which hierarchically extracts the high-level spectral features required for the classification of each pixel in the image. Spectral features were coupled with spatial-dominated information in a separate stage and fed to a logistic regression classifier as input.

In the same way, for the classification of hyperspectral data into various classes, we suggest a deep learning system. Our method, however, is based on a coherent structure that incorporates spectral and spatial information in a single stage, creating high-level spectral-spatial characteristics at the same time.

In specifically, we suggest the use of a Modified Convolutional Neural Network (CNN) that performs the operation of constructing large-level features and a Multi-Layer Perceptron (MLP) which is used for the classification of the image. The evolved framework builds spectral-spatial characteristics at once under this kind of design and, at a similar time, performs real-time predictions of the various classes in the image because of the existence of feed forward network in CNNs and MLPs.

This paper is organized into five sections: Section II gives a brief introduction and background of the convolution neural network. Section III presents the implementation methodology of the convolution neural network for hyperspectral image classification. Sections IV and V discuss the results and discussion of the image classification and conclusion.

II. CONVOLUTIONAL NEURAL NETWORK (CNN): BACKGROUND

Convolution neural network (CNN) is the most standard deep learning algorithm for image classification and image recognition problems. Along with these applications, CNN is widely used to recognize human faces and classify objects.

CNN algorithm takes the input image and extract the features in the learning phase and classify it to various class or

categories. The algorithm sees the input image as an array of a matrix ($h \times w \times d$), and it will depend on the image resolution. CNN consider the ($h \times w \times 3$) matrix for the RGB image and ($h \times w \times 1$) for the grayscale image. In principle, create the CNN model bypassing the input image series through various convolution layers with filters or kernels to extract features, pooling, fully connected layers, and finally applying the softmax function to detect the objects probabilistic value between 0 and 1. Fig. 1 shows the entire CNN follow diagram process.

A. Convolution Layer

The features existing in the input images are extracted through a convolution layer. The relationship that exists between the image's pixels is preserved by taking a small square of input data at the learning phase. The convolution is the mathematical operation given by the multiplication of an image matrix and a filter or kernel. The convolution operation of the image is shown in Fig. 2.

Consider an example of an image pixel whose matrix is 5×5 with values 0 and 1 and the filter matrix or kernel 3×3 . The convolution layer multiplies the image matrix with a filter matrix to provides the convolved feature or feature map. The same convolved operations are shown in Fig. 3 and Fig. 4.

Use the various types of filters or kernel listed in the Table I for convolution operation. These filters can also perform different mathematical calculations like sharpening of the image, blur the image and to detect the edges in the image.

B. Strides

Strides are used in convolution operation, and it determines the shift in the filter kernel matrix by certain number of pixels on the input image. For example, if stride value is 1 in convolution operation, then the filter kernel in convolution is shifted by one pixel at a time on the image matrix. If stride value is 2, then a filter kernel is shifted by two pixels at a time on the image matrix. Fig. 5 shows the operation strides by 2 pixels.

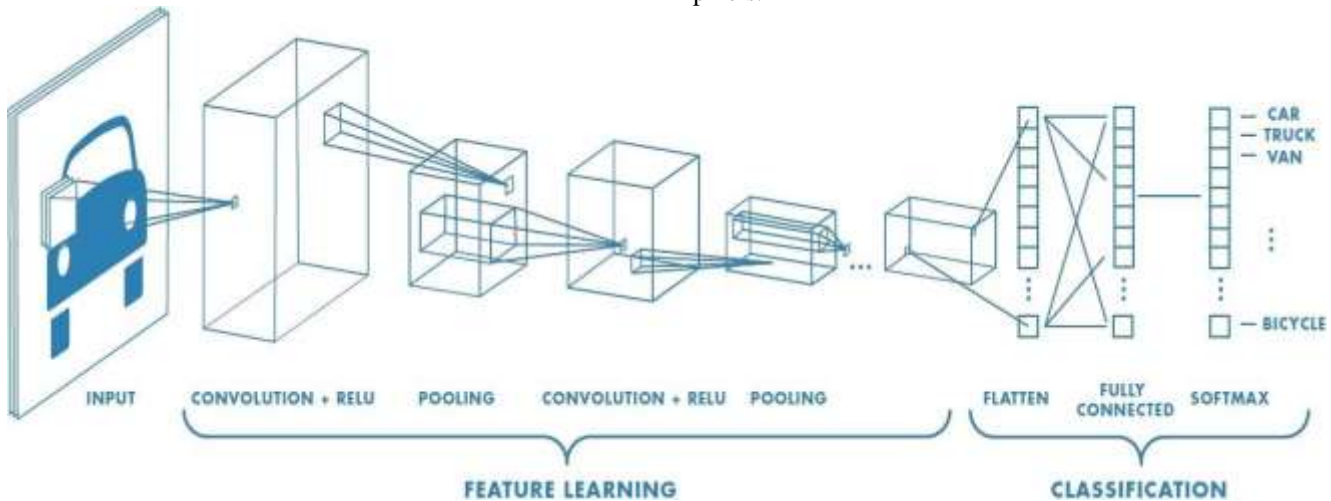


Fig. 1. CNN Flow Diagram.

- An image matrix (volume) of dimension $(h \times w \times d)$
- A filter $(f_h \times f_w \times d)$
- Outputs a volume dimension $(h - f_h + 1) \times (w - f_w + 1) \times 1$

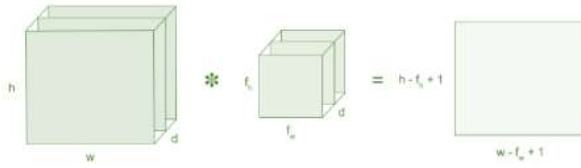


Fig. 2. Convolution Operation of the Image using Filter Kernel.

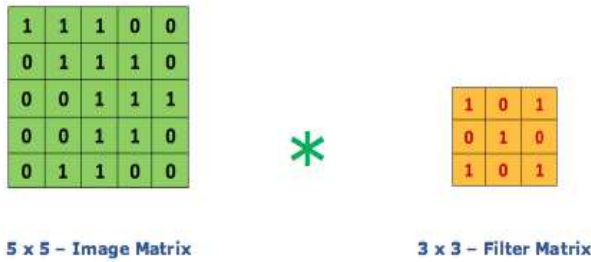


Fig. 3. Image Matrix Multiplies Kernel or Filter Matrix.

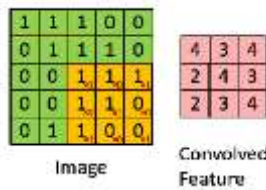


Fig. 4. 3 x 3 Output Matrix.

TABLE I. FILTERS AND ITS KERNEL FOR CONVOLUTION OPERATIONS

| Operation | Filter |
|----------------|--|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |
| Edge Detection | $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ |
| | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |
| Box blur | $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |
| Gaussian blur | $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |

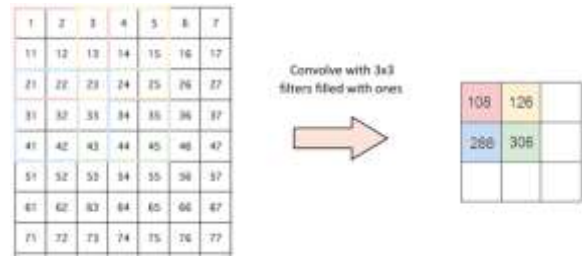


Fig. 5. Stride by 2 Pixels in Convolution Operations.

C. Padding

In most of the cases, the chosen filter does not fit to the input image. In those cases, the following options are used to fit the input images.

- Zero paddings: input image is padded with zeros to fit the input images.
- Drop a few pixels in the image to fit the input images.

D. Non Linearity (ReLU)

Rectified Linear Unit (ReLU) performs the non-linear computation in convolution network. The output of ReLU is $f(x) = \text{Max}(0, x)$.

The goal of introducing the non-linearity in ReLU is to learn the non-negative linear values in the convolution network. Fig. 6 shows the ReLU operation in CNN.

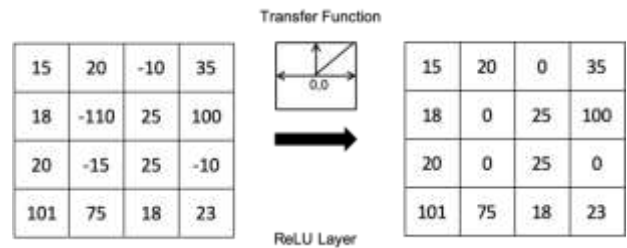


Fig. 6. ReLU Operation in CNN.

Tanh and sigmoid are the some of the non-linear functions which can be employed instead of ReLU. But the performance of ReLU is better compared to the other two non-linear operations

E. Pooling Layer

The pooling layer decreases the output pixels of the convolution layer and reduces the complexity of large images. The spatial pooling is also called subsampling or downsampling of pixels, and it decreases the spatial dimension of the image by retaining most of the information in the image. Following are three different categories of spatial pooling.

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling selects the greatest value in the convolved feature value, as shown in Fig. 7. The average pooling selects the average pixel value of all the convolved feature value. The sum pooling selects the addition of all the components in convolved feature value.

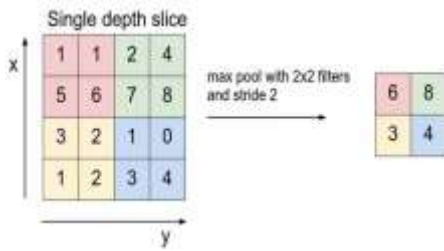


Fig. 7. Max Pooling Operation.

F. Fully Connected Layer

The two-dimensional matrix of the pooling layer is converted into a single dimension matrix. This single layer vector is feed into a fully connected layer similar to a neural network. Fig. 8 shows the operation of a fully connected layer. The output feature matrix is converted to vector as x_1, x_2, x_3 . These features are merged to construct a model using a fully connected layer and finally use softmax or sigmoid function to classify the outputs.

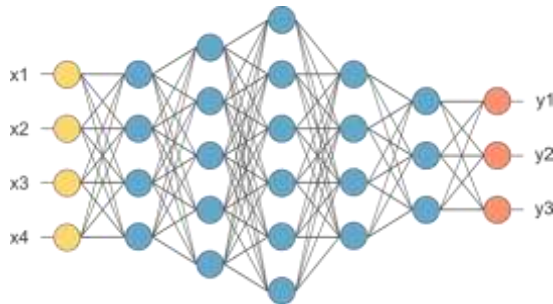


Fig. 8. Fully Connected Layer in CNN.

III. METHODOLOGY

The implementation methodology of hyperspectral image classification using the CNN approach is shown in Fig. 9. The hyperspectral image is defined as a 3-D matrix of size $h \times w \times c$, where h and w represent the image's height and width and c represents the spectral channel in the image. The hundreds of channels in the hyperspectral image enhance the calculating time and memory resources of the training and prediction process. On the other hand, using statistical analysis, it is observed that the spectral response variance is minimal for the pixel that belongs to each class.

It means for every channel, pixels with the same class labels have related spectral values, and pixels with different class labels have different spectral values. A dimensionality reduction procedure can be used, based on these properties, to decrease the dimensionality of the input data, to optimize the training and classification processes. PCA is the dimensionality reduction algorithm that reduces the hyperspectral image's

spectral dimensions without losing any image information. After dimensionality reduction, split the hyperspectral image into small patches to compatible with CNN's basic nature. Each created patch contains the spectral and spatial features of a single pixel.

More precisely, square patch of dimension $s \times s$ centred at a pixel $P_{x,y}$ is used to classify the pixel at location (x, y) on the image plane and combine spectral and spatial feature data. Let us denote the class label of the pixel at the given location as $l_{x,y}$ and patch centred at pixel $p_{x,y}$ as $w_{x,y}$. then create a dataset $D = \{w_{x,y}, l_{x,y}\}$ where $x = 1, 2, \dots, w$ and $y = 1, 2, \dots, h$. The patch $w_{x,y}$ is the 3D matrix with size $s \times s \times c$, which has spatial and spectral data for each pixel. Furthermore, this matrix $w_{x,y}$ is split into c matrixes of size $s \times s$, which are given as an input to the CNN algorithm. This CNN develops the large features that encapsulate the pixel's spectral and spatial information. The CNN architecture contains many layers, as shown in Fig. 10. The first layer in CNN structure is the convolution layer with $C_1 = 3 \times c_r$ a trainable filter of dimensions 3×3 . This layer gives C_1 matrixes of dimensions 3×3 . The second layer is also a convolution layer with $C_2 = 3 \times C_1$ trainable filter. The output of the convolution layer is given to flatten layer and fully connected layer. The flow chart for the implementation of hyperspectral image classification using a convolutional neural network is shown in Fig. 11.

IV. RESULTS AND DISCUSSIONS

The Hyperspectral image is used to simulate the proposed algorithm. Indian pines dataset is the hyperspectral image captured at north western Indiana test site using AVIRIS sensors, and its specification is shown in Table II. The overall accuracy, test loss, precision, recall and f1-score are parameters used to analyze the performance of algorithm.

In the simulation, PCA reduces the spectral bands, and 30 principal components are chosen for the classification. After dimensionality reduction, each patch contains $s \times s \times c_r$ dimensions. During the simulation, the patch size s is defined as 5 to consider the nearest 24 neighbours of each pixel. Each patch is given to the CNN architecture for classification.

The classification accuracy obtained for the 5×5 patch is 84.12%, with a simulation time of 1142.26 seconds. The screenshot of the results and the classified image is shown in Fig. 12. The test loss for the same is 47.33%. The precision, recall and f1-score of the individual class is shown in Fig. 12. The patch size s is increased to 7, 9, 11 and 13 to increase the number of neighbours for each pixel.

TABLE II. SPECIFICATION OF INDIAN PINES DATASET

| | |
|----------------|-------------------|
| Image size | 145x145 pixels |
| Spectral bands | 224 |
| Spectral range | (0.4-2.5) μ m |
| No. of Class | 16 |

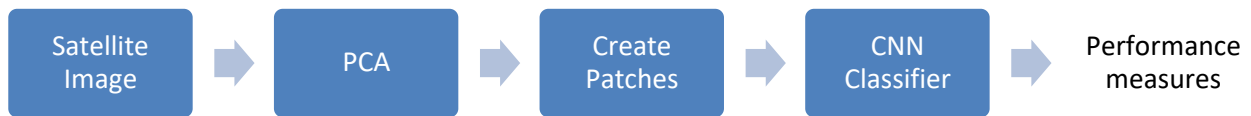


Fig. 9. Block Diagram of the Implementation of Image Classification using Convolutional Neural Networks.



Fig. 10. CNN Architecture.

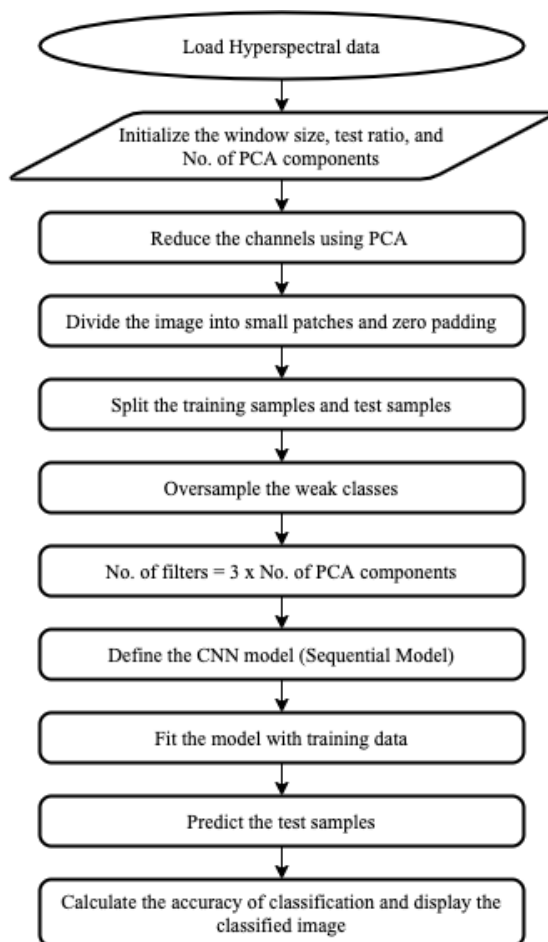


Fig. 11. Flow Chart of an Implementation of Hyperspectral Image Classification.

47.336000204086304 Test loss (%)
84.12017226219177 Test accuracy (%)

| | precision | recall | f1-score | support |
|------------------------------|-----------|--------|----------|---------|
| Alfalfa | 1.00 | 1.00 | 1.00 | 11 |
| Corn-notill | 0.73 | 0.77 | 0.75 | 357 |
| Corn-mintill | 0.74 | 0.88 | 0.80 | 208 |
| Corn | 0.77 | 0.93 | 0.85 | 59 |
| Grass-pasture | 0.96 | 0.95 | 0.95 | 121 |
| Grass-trees | 0.98 | 0.97 | 0.98 | 183 |
| Grass-pasture-mowed | 1.00 | 1.00 | 1.00 | 7 |
| Hay-windrowed | 1.00 | 1.00 | 1.00 | 120 |
| Oats | 0.83 | 1.00 | 0.91 | 5 |
| Soybean-notill | 0.68 | 0.84 | 0.75 | 243 |
| Soybean-mintill | 0.88 | 0.66 | 0.75 | 614 |
| Soybean-clean | 0.76 | 0.84 | 0.80 | 148 |
| Wheat | 1.00 | 1.00 | 1.00 | 51 |
| Woods | 0.99 | 0.97 | 0.98 | 316 |
| Buildings-Grass-Trees-Drives | 0.84 | 0.94 | 0.89 | 97 |
| Stone-Steel-Towers | 0.96 | 1.00 | 0.98 | 23 |
| accuracy | | | 0.84 | 2563 |
| macro avg | 0.88 | 0.92 | 0.90 | 2563 |
| weighted avg | 0.85 | 0.84 | 0.84 | 2563 |

Fig. 12. Screenshot of CNN Classification with 5 x 5 Patch Size.

Increasing the patch size enhance the classification accuracy and increases the simulation time. The results of CNN classification with different patch size is shown in Fig. 13. The accuracy and computation time of CNN classification is given in Table III. From the table, it can be seen that CNN achieves the highest classification accuracy and computation time for the patch size 13 x 13. No further improvements are observed in classification accuracy for the patch size of more than 13, and in fact, the accuracy of the classifier deteriorates and increases the computational resources. The proposed CNN algorithm for the classification of the hyperspectral image is compared with the Support vector machine classifier as shown in Table IV. Support vector machine uses the spectral features for classification and spatial features are add to SVM output by using guided image filter and bilateral filter.

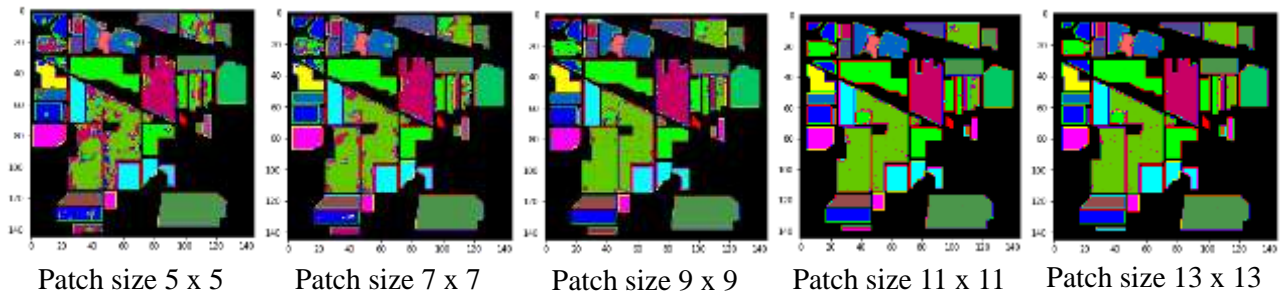


Fig. 13. Results of CNN Classification with different Patch Size.

TABLE III. ACCURACY AND COMPUTATION TIME OF CNN CLASSIFICATION WITH DIFFERENT PATCH SIZE

| Patch Size | 5 x 5 | 7 x 7 | 9 x 9 | 11 x 11 | 13 x 13 |
|------------------------|---------|----------|----------|----------|----------|
| Overall Accuracy(%) | 84.12 | 91.10 | 95.00 | 97.42 | 98.28 |
| Computation time (sec) | 1142.26 | 2111.512 | 3047.524 | 4263.925 | 5002.114 |

TABLE IV. COMPARISON OF SVM AND CNN CLASSIFICATION METHODS

| Methods | SVM [8] | SVM with PCA [8] | SVM with LDA [9] | SVM with GIF [8] | SVM with PCA & GIF [8] | SVM with LDA and GIF [9] | SVM with BF [9] | SVM with PCA & BF [9] | SVM with LDA and BF [9] | CNN |
|------------------------|---------|------------------|------------------|------------------|------------------------|--------------------------|-----------------|-----------------------|-------------------------|----------|
| Overall Accuracy (%) | 91.28 | 88.1 | 86.53 | 99.04 | 98.99 | 96.37 | 98.89 | 98.93 | 96.53 | 98.28 |
| Computation Time (sec) | 4117.7 | 568.11 | 423.58 | 4118.15 | 568.25 | 424.1 | 4118.03 | 568.32 | 424.08 | 5002.114 |

Along with SVM, the dimensions of the spectral features are reduced using PCA and LDA. The SVM method along with guided image filter and bilateral filter increases the classification accuracy and SVM method along with PCA and LDA decreases the computational time. CNN algorithm combines both spectral and spatial features at the same time without using guided image filter and bilateral filter. CNN algorithm archives the highest classification accuracy compared to the SVM classification accuracy with increasing computation time. SVM algorithm consumes less computational time compared with CNN algorithm.

V. CONCLUSION

The hyperspectral image is classified using a convolutional neural network. Both spectral and spatial features are considered to classify the image. Spatial features are included in the classification to improve the classification. CNN archives the highest classification accuracy of 98.28% compared to support vector machine classifier and other methods. CNN accuracy depends on the patch size considered for the classification. Patch size indicates the number of spatial features considered for classification. It is observed that the patch size of 13 x 13 is enough to achieve the highest accuracy. CNN consumes more computation time for training and testing compared with other classifiers. The proposed method avoids the usage of an edge-preserving filter to incorporate spatial features into the classification.

REFERENCES

- [1] S.Li, W.Song, L.Fang, Y.Chen, P.Ghamisi and J.A.Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57(9), pp. 6690-6709, 2019.
- [2] D. Saqui, J. H. Saito, L. A. D. C. Jorge, E. J. Ferreira, D. C. Lima and J. P. Herrera, "Methodology for Band Selection of Hyperspectral Images Using Genetic Algorithms and Gaussian Maximum Likelihood Classifier," *International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, 2016, pp. 733-738.
- [3] B. Tu, J. Wang, X. Kang, G. Zhang, X. Ou and L. Guo, "KNN-Based Representation of Superpixels for Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4032-4047, Nov. 2018.
- [4] W. Li, S. Prasad, E. W. Tramel, J. E. Fowler and Q. Du, "Decision fusion for hyperspectral image classification based on minimum-distance classifiers in the wavelet domain," *IEEE China Summit & International Conference on Signal and Information Processing*, Xi'an, 2014, pp. 162-165.
- [5] S. Bajpai, H. V. Singh and N. R. Kidwai, "Feature extraction & classification of hyperspectral images using singular spectrum analysis & multinomial logistic regression classifiers," *2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, Aligarh, 2017, pp. 97-100.
- [6] Q. Sun, X. Liu and M. Fu, "Classification of hyperspectral image based on principal component analysis and deep learning," *International Conference on Electronics Information and Emergency Communication*, 2017, pp. 356-359.
- [7] S. Zhong, C. Chang and Y. Zhang, "Iterative Support Vector Machine for Hyperspectral Image Classification," *International Conference on Image Processing*, 2018, pp. 3309-3312.

- [8] Shambulinga M and Sadashivappa G., "Hyperspectral Image Classification using Support Vector Machine with Guided Image Filter" International Journal of Advanced Computer Science and Applications, vol. 10(10), 2019.
- [9] Shambulinga M and G. Sadashivappa, "Supervised Hyperspectral Image Classification using SVM and Linear Discriminant Analysis" International Journal of Advanced Computer Science and Applications, Vol. 11(10), 2020.
- [10] M.S.Aydemir and G.Bilgin, "Semisupervised Hyperspectral Image Classification Using Deep Features," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12(9), pp. 3615-3622, 2019.
- [11] I.Bidari, S.Chickerur, H.Ranmale, S.Talawar, H.Ramadurg and R.Talikoti, "Hyperspectral Imagery Classification Using Deep Learning," *Conference on Smart Trends in Systems, Security and Sustainability*, 2020, pp. 672-676.
- [12] S.Liu, Q.Shi and L.Zhang, "Few-Shot Hyperspectral Image Classification With Unknown Classes Using Multitask Deep Learning," in *IEEE Transactions on Geoscience and Remote Sensing*, 2020.