Multispectral Image Analysis Using Deep Neural Networks

Arun D. Kulkarni
Computer Science Department, The University of Texas, Tyler, TX 75799, USA

***Abstract*—Multispectral image classification plays a crucial role in remote sensing applications such as land cover mapping, agricultural monitoring, and environmental surveillance. Traditional classification techniques, including the Maximum Likelihood Classifier (MLC), Support Vector Machine (SVM), Decision Tree (DT), and Multi-Layer Perceptron (MLP), often struggle with the complexity and high dimensionality of multispectral data. Recent advances in deep learning have revolutionized the field of remote sensing by enabling the extraction of high-level, abstract features from raw input data. In this paper, we explore the application of Deep Neural Networks (DNNs) for pixel-wise classification in multispectral imagery. DNNs are capable of learning informative and hierarchical representations, which have demonstrated significant success in a wide range of computer vision tasks. We propose and implement a simple DNN architecture consisting of six layers: an input layer (representing reflectance values across spectral bands), a fully connected layer, a batch normalization layer, a ReLU activation layer, another fully connected layer, and a final SoftMax output layer for classification. Each pixel is represented by a vector of spectral reflectance values. We evaluated our model using two Landsat scenes, one from the New Orleans area and the other from the Mississippi River bottomland area. The proposed DNN achieved classification accuracy of 97.44% and 95.74%, respectively, on these datasets, demonstrating the effectiveness of deep learning for multispectral image classification.**

Keywords—Remote sensing; classification; deep neural networks; Landsat scene

I. Introduction

Multispectral imaging has become an essential tool in the field of remote sensing, enabling the acquisition of detailed information about the Earth's surface by capturing data across multiple distinct spectral bands. Unlike conventional RGB imagery, which captures only three visible bands (red, green, and blue), multispectral images span a broader range of the electromagnetic spectrum, including the visible, near-infrared (NIR), and shortwave infrared (SWIR) regions, providing richer spectral detail. This additional spectral information enhances the ability to distinguish between materials and land cover types, making multispectral imagery invaluable for a variety of applications, such as agricultural monitoring, land use classification, forestry, environmental change detection, and disaster assessment.

In remote sensing, pixel-based classification is one of the preferred methods for analyzing multispectral images. However, classifying multispectral images presents several challenges. The increased dimensionality of the data, combined with high inter-class similarity and intra-class variability, makes it difficult for traditional machine learning (ML) algorithms to achieve high classification accuracy. Common ML techniques used for pixel-based classification of multispectral images include the maximum likelihood classifier (MLC), decision tree (DT), support vector machine (SVM), random forest (RF), k-nearest neighbors (k-NN), and neural network (NN) models. In pixel-based classification, each pixel is represented as a vector of reflectance values across multiple spectral bands. A few sample pixels are selected from homogeneous areas representing distinct ground categories to form the training dataset. The classifier model is trained using this data, and once trained, the model is applied to classify the entire image scene. In the late 1980s, neural network (NN) models such as the multi-layer perceptron model with backpropagation learning were used for pixel-based classification of multispectral imagery. One advantage of NN models is that they are more efficient compared to conventional machine learning models, such as maximum likelihood classification (MLC).

In recent years, deep learning has revolutionized the field of image analysis by introducing models capable of learning complex, non-linear representations of data directly from raw inputs. A major limitation of early NN models using gradient descent with sigmoid activation functions was the vanishing gradient problem, where saturation of activation functions led to slow convergence during training. To mitigate this issue, deep learning (DL) models often employ entropy-based loss functions along with rectified linear unit (ReLU) activations in the output layer. Overfitting commonly occurs when the dataset is small. To address this, various regularization techniques such as dropout and bagging are employed. DL models can be trained on large datasets and can achieve high classification accuracy. DL-based pixel classification for multispectral images involves designing architectures that support pixel-wise data representation and classification. By adopting DL techniques, it becomes possible to extract more abstract and robust feature representations, thereby improving classification performance. DL models can learn informative representations of raw input data through multiple levels of abstraction. These features have achieved notable success across many computer vision tasks. Among deep learning architectures, convolutional neural networks (CNNs) have demonstrated exceptional performance in tasks such as image classification, object detection, and segmentation. In the context of multispectral image classification, deep learning offers several key advantages: it eliminates the need for manual feature extraction, captures both spectral and spatial dependencies, and scales effectively with large and high-dimensional datasets. Furthermore, advanced architectures such as 3D CNNs, recurrent neural networks (RNNs), and hybrid models have been proposed to better exploit the unique characteristics of multispectral and hyperspectral data. Transfer learning and data augmentation techniques have also been utilized to address the common challenge of limited labeled datasets in remote sensing. These innovations have led to significant improvements in classification performance, establishing deep learning as a leading approach in this domain.

This study focuses on the classification of multispectral images using deep learning techniques. We propose a simple one-dimensional deep neural network (DNN) architecture to classify pixels in multispectral imagery. To validate our approach, we implemented a six-layer DNN using the MATLAB Deep Learning Toolbox. We analyzed two Landsat scenes: one from the New Orleans area and another from the Mississippi River bottomland area. Training data were extracted by displaying the scenes on a monitor and selecting small homogeneous regions representing distinct ground categories. For the Mississippi scene, four training areas were selected, each corresponding to a different category. For the New Orleans scene, three training areas were selected, each representing a distinct category. The remainder of the paper is organized as follows: Section II reviews related work; Section III describes the proposed approach; Section IV presents the experiments and results; and Section V concludes the study.

II. Related Work

Earlier well-known machine learning (ML) algorithms, such as the maximum likelihood classifier (MLC), support vector machine (SVM), decision tree (DT), random forest (RF) and k-Nearest Neighbor (k-NN) were used for classifying pixels in multispectral images. The MLC algorithm is one of the most well-known algorithms. It assumes the normal distribution for reflectance values and calculates the mean vector and covariance matrix for each class using training data. The classifier uses Bayes' rule to calculate posterior probabilities and assigns a pixel to the class with the highest posterior probability. The SVM is a binary classifier that assigns a sample to one of the two linearly separable classes. In the SVM algorithm, two hyper-planes are selected so as not only to maximize the distance between the two classes but also not to include any points between them. The SVM algorithm is extended to nonlinearly separable classes by mapping samples to a higher-dimensional feature space. The SVM algorithm is appealing for Landsat data analysis because of its ability to successfully handle small datasets, often producing higher classification accuracy than traditional methods. Moumtrakis et al. [1] have provided a review of usage of SVM in remote sensing. Kulkarni and Shrestha [2] have used DT to classify Landsat imagery. Neural networks are preferred for classification because of their parallel processing capabilities as well as learning and decision-making abilities. Several studies aimed at evaluating the performance of neural networks in comparison with traditional statistical methods for remote sensing applications are available. The rapid uptake of neural approaches in remote sensing is due mainly to their widely demonstrated ability to: 1) perform more accurately than other techniques, such as statistical classifiers, particularly when the feature space is complex and the source data has diverse statistical distributions; 2) perform more rapidly than other techniques, such as different sensors; 3) incorporate a priori knowledge and realistic physical constraints into the analysis, incorporate different types of data (including those from different sensors) into the analysis, thus facilitating synergistic studies. Neural networks with learning algorithms such as backpropagation (BP) can learn from training samples and are used in Landsat data analysis. Fuzzy-neural systems that combine neural networks and fuzzy logic have been used to classify pixels in remote sensing imagery.

Deep learning has achieved impressive performance with applications such as image classification and object detection. Deep learning is the fastest growing trend in big data analysis. Deep learning is characterized by neural networks that usually have more than two hidden layers. Zhang, et al. [3] provide a tutorial on the use of deep learning in remote sensing. Remote sensing data are often multispectral and face big data challenges. Deep learning provides more methodologies to train deep NN architectures, such as autoencoder models Convolution Neural Networks (CNNs). Earlier well-known CNN models include Alex Net, VGG Net, and Reset. Feature representations learned by CNNs are highly effective in object recognition. Zhu et al. [4] provide an overview of deep learning techniques for remote sensing. They discuss various architectures, data sets, and performance metrics commonly used in multispectral image classification. Many comprehensive survey articles on deep learning applied to remote sensing are available in the literature [5-9]. CNNs have become instrumental in extracting spatial and spectral features from satellite and aerial imagery for a variety of applications. CNNs have demonstrated significant performance improvements in image classification, object detection, and semantic segmentation, making them highly suitable for remote sensing applications. In remote sensing, CNNs have been used for applications such as land cover and land use classification, object detection, semantic segmentation, change detection, and agriculture monitoring. Zhu et al. [4], in their review article, summarize the basic principles of deep learning and its research progress and typical applications in remote sensing. Their article focuses on the research status of deep learning in remote sensing image classification and object detection. Deep learning techniques, especially CNNs, have revolutionized the field of remote sensing, particularly pixel-based classification tasks. The DL approaches for pixel-based multispectral image classification involve 1) preparing an input vector representing a pixel in the scene, 2) developing and training a hierarchical DL model, and 3) classifying the entire scene. In the first step, the input vector could represent spectral features, spatial features, or spectral and spatial features. Hidden layers in the deep network structure are designed to learn feature representations of the input data. The last classification stage involves decision making. Utilizing learned features from previous stages. In hard classification, the classifier directly outputs an integer number as the label of each sample. Hu et al. [10] train a simple one-dimensional CNN that contains five layers: an input layer, a convolution layer, a max-pooling layer, a fully connected layer, and an output layer that was used to classify pixels in hyper-spectral imagery. Feature vectors representing the spectral signatures were used as an input. Their results demonstrate that their proposed method can achieve better classification performance compared with SVM-based and conventional deep NN-based classifiers. Maggiori et al. [11] proposed an end-to-end framework for the dense, pixelwise classification of satellite imagery with convolutional neural networks (CNNs). In their framework, CNNs are directly trained to produce classification maps out of the input images. Kussul et al. [12] proposed a multi-level deep learning architecture that targets land cover and crop type classification from multitemporal and multisource satellite imagery. Their results show that architecture with an ensemble of CNNs outperforms the one with a multi-layer perception. Kulkarni [13] proposed an architecture to classify pixels based on their spectral signatures using CNN models. In the proposed framework, each feature vector was mapped to a two-dimensional image. The mapped images were then classified to respective classes. Li et al. [14] provide a survey of deep learning techniques for remote sensing image mapping. They elucidate four predominant strategies for matching: arar-based, feature-based, regression-based, and unsupervised learning. Lian et al. [15] provide an overview of advances in deep learning-based spatiotemporal fusion methods for remote sensing images.

U-Net is one of the types of deep network models that is specifically built for image segmentation, which was introduced in 2015 [16]. Long et al. [17] have shown that CNN trained end-to-end, pixel-to-pixel, exceed the state-of-the-art in semantic segmentation. They define and detail the space of fully convolutional networks and explain their application in spatially dense prediction tasks. Solórzano et al [18] use U-net for land cover classification. Sahin et al. [19] trained the U-Net with images from the cotyledon emergence through to the subsequent growth stages and evaluated images of crops in the last stage of growth. The results of their experiment demonstrate that the proposed method, with suitable input, can be used for crop and weed detection or segmentation. Wiratama et al. [20] proposed a change detection algorithm on multi-spectral images based on a feature-level U-net. Their algorithm gave a better performance compared to the existing change detection algorithms. Bai et al. [21] provide a review of deep learning methods for change detection in remote sensing images.

III. Proposed Approach

A. Data Collection and Pre-processing

In this study, we used two Landsat scenes obtained Operational Land Imager (OLI) representing the New Orleans area located at Latitude N 30.55580 and Longitude W 89.92440. The second scene represents the Mississippi River bottomland area located at Latitude N 34.65710 and Longitude W 90.40900. We used spectral bands 2, 3, 4, 5, and 7 as these bands showed the maximum variance. The training set data were obtained by displaying the scenes and selecting homogeneous areas representing the classes. The training set data were used to train the DNN models. For the New Orleans scene, the training set contained a total of 1200 pixels, 400 pixels, and for the Mississippi River bottomland scene, the total number of training data was 3600 pixels, 900 from each class. Each pixel was represented by a vector of size 5. Seventy percent random samples were used to train the DNN model, and 30 percent random samples were used for testing. The reflectance values were normalized between 0 and 10.

B. DNN Architecture

The classification model used in this study was a simple Deep Neural Network (DNN) built in MATLAB, comprising six layers. The network was designed to process the multispectral feature vectors and classify the pixels into distinct categories based on their spectral characteristics. The architecture of the model is shown in Fig. 1. The layers in the model are as below.

1) Feature input layer: The first layer of the network was the input layer, where each pixel's feature vector, consisting of five spectral reflectance values (one for each band), was fed into the network. This layer was designed to accept data with a dimensionality corresponding to the number of spectral bands used.

2) Fully connected layer: The second layer of the network was a fully connected (dense) layer, which applied a set of learnable weights to the input features. This layer allowed the network to begin learning the complex relationships between the spectral bands.

3) Batch normalization layer: To enhance the training stability and speed up convergence, a batch normalization layer was applied. This layer normalizes the output of the fully connected layer, ensuring that the data fed into the next layer is centered and scaled appropriately. This normalization helped reduce internal covariate shift and improved model generalization.

4) ReLU (Rectified Linear Unit) activation layer: The next layer was a ReLU activation layer. ReLU introduces non-linearity into the model, allowing it to learn more complex features from the input data. The ReLU function was applied elementwise to the outputs of the batch normalization layer, ensuring that only positive values passed through to subsequent layers. ReLU later helps mitigate the vanishing gradient problem

5) Fully connected layer: The fifth layer was another fully connected layer, which aimed to further abstract and combine features learned from the previous layers. This layer had a greater number of neurons than the first fully connected layer, enabling the model to capture more intricate relationships in the data.

6) SoftMax layer: The fully connected layer was followed by a SoftMax layer. SoftMax layer takes a vector of raw scores (often called logits) from the final hidden layer and converts them into a probability distribution over multiple classes. The scores are exponentiated and then normalized so that they sum to 1.

The model was trained using the training data collected from the homogeneous regions. During training, the model optimized its weights using backpropagation and stochastic gradient descent (SGD). The loss function used for training was categorical cross-entropy, as the goal was to classify each pixel into one of several predefined classes. A portion of the training data was set aside for validation to monitor the model's performance during training and to prevent overfitting. Training was performed for a predetermined number of epochs, with the learning rate adjusted dynamically to ensure efficient convergence. Early stopping was used to prevent overfitting, where training was halted once the validation accuracy plateaued over several epochs. To evaluate the performance of the trained model, several metrics were computed, including overall accuracy, precision, recall, F1-score, and ROC curves.

The predicted categories were compared to the true categories in test sets.

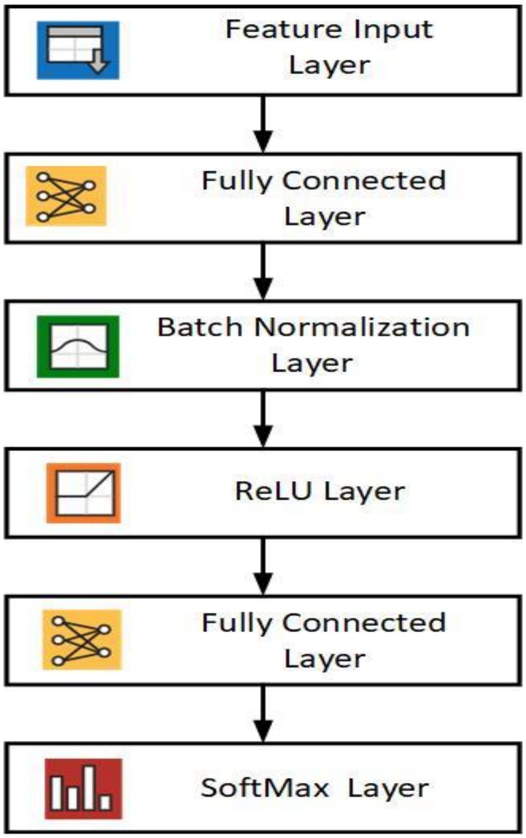

Fig. 1. Deep Neural Network (DNN) architecture.

IV. Implementation and Results

In this research work, we implemented the DNN model using MATLAB scripts from deep learning and analyzed two Landsat scenes.

A. *Example 1: New Orleans Scene*

The scene was obtained by Landsat-8 OLI, representing the New Orleans area. The path and row numbers for the scenes are 22 and 39, respectively. To generate the training data, we considered a scene of the size 512 by 512 pixels. Three small homogeneous areas were selected as training sets that represent three classes: water, land, and vegetation. The training data contains 1200 samples, 400 from each class. We selected bands 2, 3, 4, 5, and 7. The spectral signatures obtained from mean vectors of the classes are shown in Fig. 2. The scatter plot for the dataset is shown in Fig. 3. We used 70 percent randomly selected samples for training and 30 percent for validation. With the DNN, we obtained the overall accuracy of 97.5 percent. The learning progress curve for the DNN is shown in Fig. 4. The confusion matrix is shown in Fig. 5. The ROC curves and classified output are shown in Fig. 6 and 7. Table I shows the evaluation metric with Recall, Precision, F-Score, accuracy, and specificity for the classes in the data set.

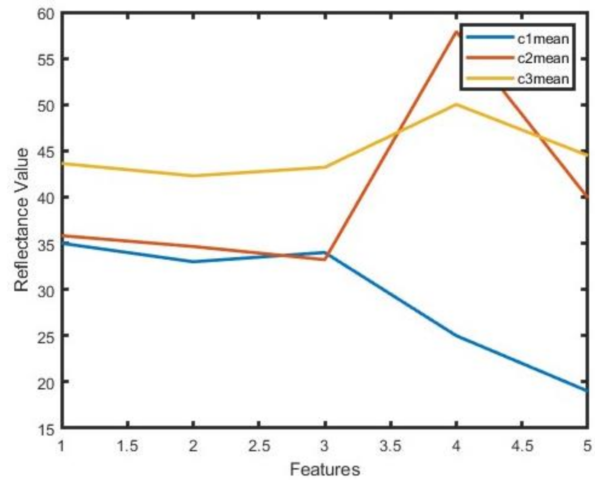

Fig. 2. Spectral signatures (New Orleans scene).

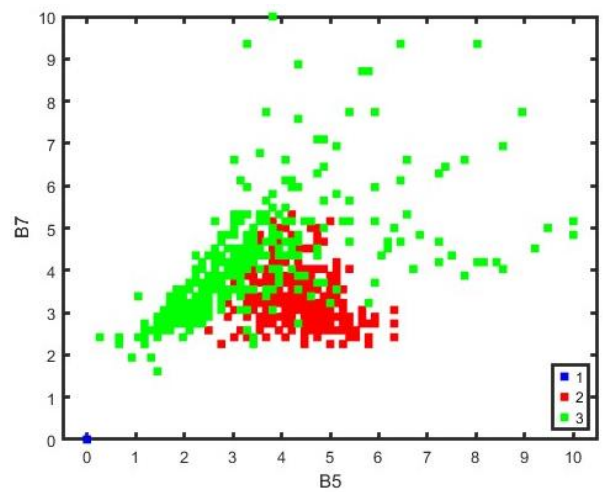

Fig. 3. Scatter plot (New Orleans scene).

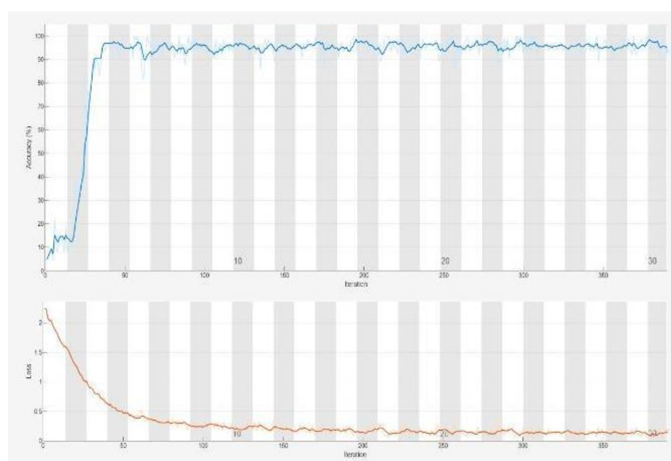

Fig. 4. Training progress (New Orleans scene).

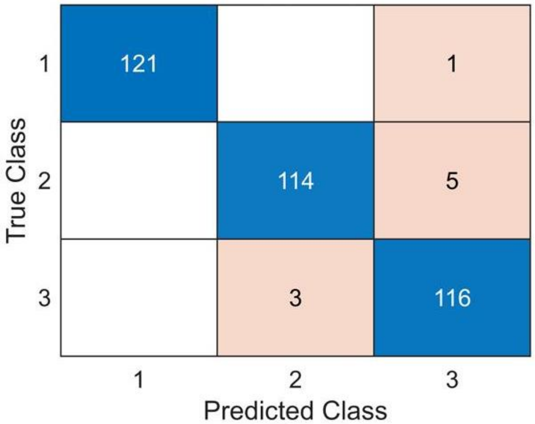

Fig. 5. Confusion matrix (New Orlean scene).

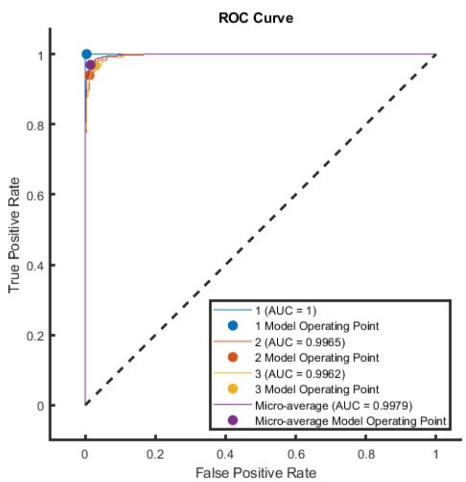

Fig. 6. ROC Curve (New Orleans scene).

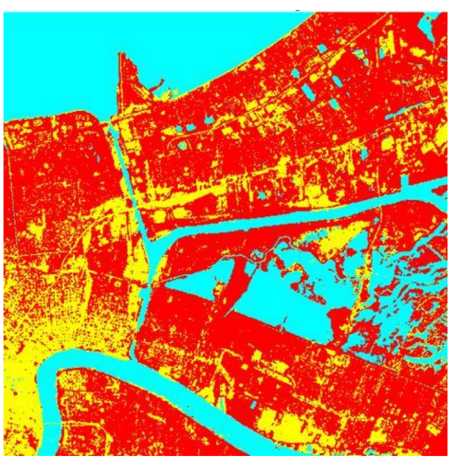

Fig. 7. Classified output (New Orleans scene).

TABLE I. EVALUATION METRICS (NEW ORLEANS SCENE)

Class	Recall	Precision	F-Score	Accuracy	Spec
1	0.9918	1.0	0.9959	0.9972	1.0
2	0.9580	0.9744	0.9661	0.9778	0.9876
3	0.9748	0.9508	0.9627	0.9750	0.9751

B. Example 2: Mississippi Scene

The second scene is of the Mississippi bottomland. The path and row numbers are 23 and 36, respectively. We considered the scene of size 512 by 512 pixels. Training and test data were acquired in the same manner as the first scene. Classes of water, soil, forest, and agriculture were chosen. The training data contains 3600 samples, 900 from each class. We selected bands 2, 3, 4, 5, and 7. The spectral signatures obtained from mean vectors of the classes are shown in Fig. 8. Fig. 9 shows the scatter plot for the dataset. We used 70 percent randomly selected samples for training and 30 percent for validation. With the DNN, we obtained the overall accuracy of 95.74 percent. The learning progress curve for the DNN is shown in Fig. 10. The confusion matrix is shown in Fig. 11. The ROC curves and classified output are shown in Fig. 12 and 13, respectively. Table II shows the classifier evaluation metrics.

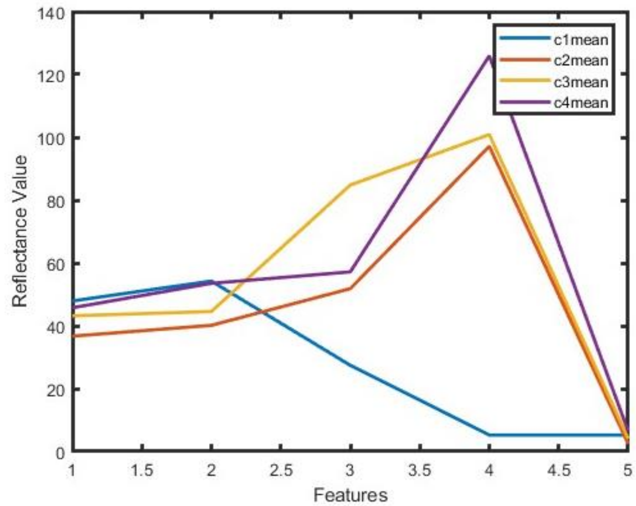

Fig. 8. Spectral signatures (Mississippi scene).

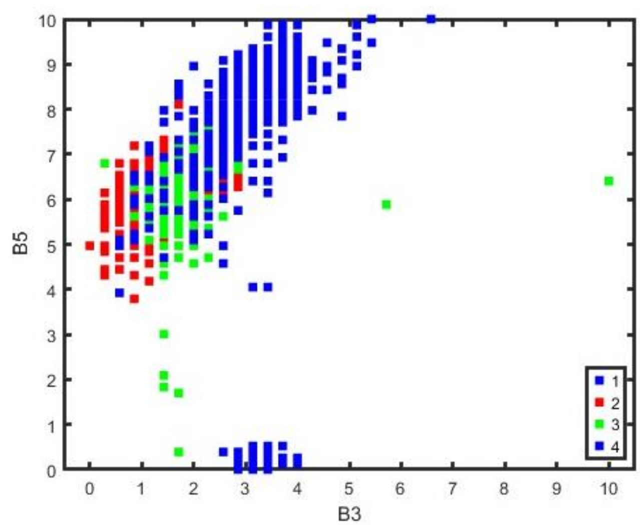

Fig. 9. Scatter plot (Mississippi scene).

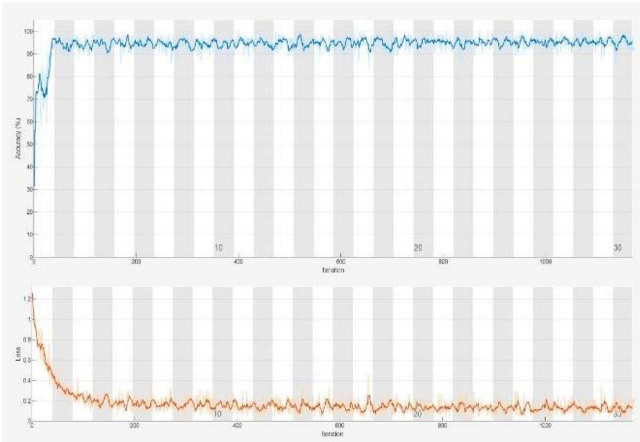

Fig. 10. Learning progress (Mississippi scene).

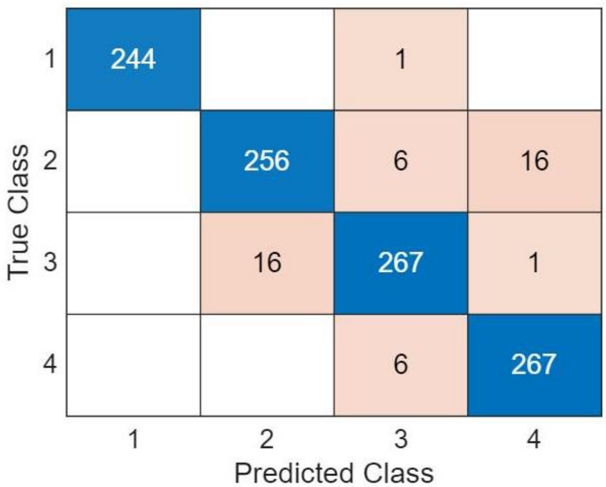

Fig. 11. Confusion matrix (Mississippi scene).

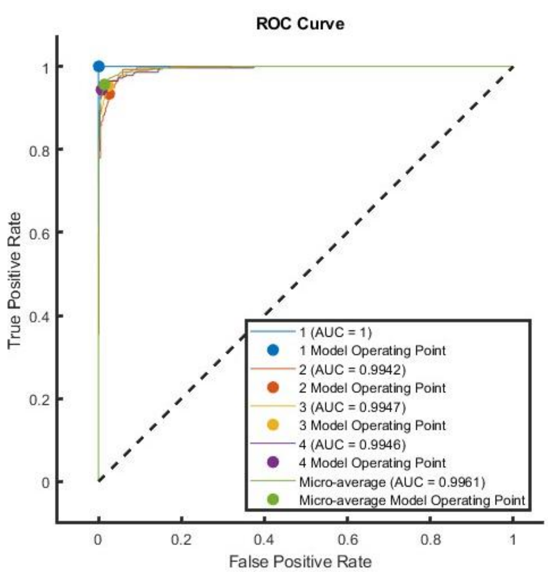

Fig. 12. ROC Curve (Mississippi scene).

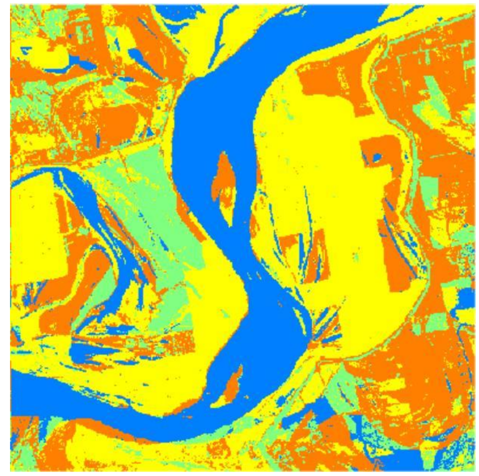

Fig. 13. Classified output (Mississippi scene).

TABLE II. EVALUATION METRICS (MISSISSIPPI SCENE)

Class	Recall	Precision	F-Score	Acc	Spec
1	0.9959	1	0.9980	0.9991	1
2	0.9209	0.9412	0.9309	0.9648	0.9800
3	0.9401	0.9536	0.9468	0.9722	0.9837
4	0.9780	0.9401	0.9587	0.9787	0.9789

V. CONCLUSION

In this paper, we propose and implement a simple deep neural network (DNN) model to classify pixels in multispectral images. The model was developed using MATLAB scripts. We analyzed two Landsat-8 scenes: one from the New Orleans area and another from the Mississippi River bottomland. We chose the Landsat scenes that are available in the public domain [22]. Training data were generated by selecting small, homogeneous areas within the displayed scenes. Feature vectors representing reflectance values were used to train the DNN model. The New Orleans scene included three categories, while the Mississippi River bottomland scene included four. The DNN model achieved classification accuracies of 97.44% and 95.74% for the New Orleans and Mississippi River bottomland scenes, respectively. Our experiments demonstrate that DNN models provide a powerful alternative to conventional machine learning techniques for analyzing remote sensing data. Although this work focuses solely on spectral signatures and does not incorporate spatial correlation, we believe that combining spatial and spectral techniques could further enhance DNN-based classification. Future work will involve extending the algorithm to hyperspectral images with a greater number of spectral bands and evaluating the model using standard benchmark datasets. The DNN model can be easily extended for hyperspectral data by increasing the size of the feature input and fully connected layers. CNN-based hybrid networks have been used for pixel-based classification. These networks are complex and time consuming. We would like to compare the present DNN model with CNN-based hybrid models for performance.

ACKNOWLEDGMENT

The author is thankful to the anonymous reviewers for their valuable suggestions, which have helped improve the quality of the paper.

REFERENCES

[1] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," ISPRS Journal of Photogrammetry and Remote Sensing, 2011, vol. 66, no. 3, pp. 247-259.

[2] A. D. Kulkarni and A. Shrestha. "Multispectral image analysis using decision trees", International Journal of Advanced Computer Science and Applications, 2017, vol. 8, no. 6, pp 11-18.
doi: 10.14569/IJACSA.2017.080602.

[3] L. Zhang, L. Zhang and B. Du, "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art," IEEE Geoscience and Remote Sensing Magazine, 2016, vol. 4, no. 2, pp. 22-40, doi: 10.1109/MGRS.2016.2540798.

[4] X. X. Zhu et al., "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," IEEE Geoscience and Remote Sensing Magazine, 2017, vol. 5, no. 4, pp. 8-36, doi: 10.1109/MGRS.2017.2762307

[5] M. Zhu,Y. He, and Q. He. "A Review of Research on Deep Learning" Remote Sensing Application. International Journal of Geosciences, 2019 vol. 10, 1-11. doi: 10.4236/ijg.2019.101001.

[6] J. S. Song, S. Gao, Y. Zhu, and C. Ma. "A survey of remote sensing image classification based on CNNs". Big Earth Data, 2019, vol. 3. No. 3. pp. 232–254. https://doi.org/10.1080/20964471.2019.1657720

[7] E. M. Rewhel, J. Q. Li, A. A. Hamed. Et al. 2023. Deep Learning Methods Used in Remote Sensing Images: A Review. Journal of Environmental & Earth Sciences. 2023, vol. 5, no. 1, pp 33-64. doi.org/10.30564/jees.

[8] X. Wu, D. Hong, and J. Chanussot, "Convolutional Neural Networks for Multimodal Remote Sensing Data Classification," IEEE Transactions on Geoscience and Remote Sensing, 2022, vol. 60, pp. 1-10. doi: https://10.1109/TGRS.2021.3124913.

[9] S. Paheding, A. Saleem, M. F. H. Siddiqui, et al. "Advancing horizons in remote sensing: a comprehensive survey of deep learning models and applications in image classification and beyond". Neural Comput & Applic, 2024 vol. 36, 16727–16767. https://doi.org/10.1007/s00521-024-10165-7.

[10] W. Hu, Y. Huang, L. Wei, F. Zhang. H. Li. "Deep Convolutional Neural Networks for Hyperspectral Image Classification". Journal of Sensors. Volume 2015, no.1 1 258619. https://doi.org/10.1155/2015/258619.

[11] E. Maggiori et al. "Convolutional neural networks for large-scale remote-sensing image classification." IEEE Transactions on geoscience and remote sensing vol. 2016, 55, no.2, pp 645-657.

[12] N. Kussul, M. Lavreniuk, S. Skakun and A. Shelestov, "Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data," IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 5, pp. 778-782, May 2017, doi: 10.1109/LGRS.2017.2681128.

[13] A. D. Kulkarni. "Multispectral Image Analysis Using Convolution Neural Networks". International Journal of Advanced Computer Science and Applications, vol. 14, no. 10, 2023, pp. 13-19. doi: 10.14569/IJACSA.2023.0141002

[14] R. Li, S. Zheng, C. Duan, L. Wang, and C. Zhang. (2025). "Land cover classification from remote sensing images based on multi-scale fully convolutional network". Geo-Spatial Information Science, 2025, vol. 25, no. 2, pp. 278–294. https://doi.org/10.1080/10095020.2021.2017237

[15] Z. Lian, Y. Zhan, W. Zhang, Z. Wang, W. Liu, X. Huang. "Recent Advances in Deep Learning-Based Spatiotemporal Fusion Methods for Remote Sensing Images". Sensors (Basel). 2025 vol. 25, no. 4, pp1093. doi: 10.3390/s25041093. PMID: 40006322; PMCID: PMC11859923.

[16] O. Ronneberger, P. Fischer, T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[17] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.

[18] J. V. Solórzano, J. F. Mas, Y. Gao, and J. A. Gallardo-Cruz. "Land Use Land Cover Classification with U-Net: Advantages of Combining Sentinel-1 and Sentinel-2 Imagery". Remote Sens. 2021, vol. 13, no.18, 3600; https://doi.org/10.3390/rs13183600

[19] H. M. Sahin, T. Miftahushudur, B. Grieve, H. Yin. "Segmentation of weeds and crops using multispectral imaging and CRF-enhanced U-Net". Computers and Electronics in Agriculture, vol. 211, August 2023, pp. 107956. https://doi.org/10.1016/j.compag.2023.107956

[20] W. Wiratama, J. Lee and D. Sim, "Change Detection on Multi-Spectral Images Based on Feature-level U-Net," IEEE Access, 2020, vol. 8, pp. 12279-12289. doi: 10.1109/ACCESS.2020.2964798.

[21] T. Bai., L. Wang, D. Yin, K. Sun, Y. Chen, W. Li, and D. Li. Deep learning for change detection in remote sensing: a review. Geo-Spatial Information Science, 2022, vol. 26, no. 3, pp. 262–288. https://doi.org/10.1080/10095020.2022.2085633.

[22] USGS Landsat Data: https://www.usgs.gov/landsat-missions/landsat-data-access.