International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 7, Issue 1, 357-362, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I1P157

Original Article

Hyperspectral Image Classification using Principal Component Analysis and Vision Transformer

Pashikanti Siddartha[1], Parwathala Varun Gaurav[2], Shaik Shaad Mehraj[3], Divila Sricharan[4], S. Vijitha[5]
[1,2,3,4]Department of Artificial Intelligence, Anurag University , Hyderabad,India.
[5]Assistant Professor, Department of Artificial Intelligence, Anurag University, Hyderabad, India.

Received On: 12/02/2026 ***Revised On: 11/03/2026*** ***Accepted On: 14/03/2026*** ***Published On: 17/03/2026***

Abstract - *Hyperspectral image classification is essential in remote sensing applications, aiming to accurately categorize land cover or material depicted in hyperspectral data. This paper makes use of Vision transformer model with Principal component analysis as a dimensionality reduction technique. The Vision Transformer model utilizes self-attention mechanisms to capture spatial dependencies among image patches unlike traditional CNN layers. Experiments were conducted on standard hyperspectral datasets including Indian Pines, University of Pavia for training the model.*

Keywords - *Hyperspectral Image Classification, Principal Component Analysis, Vision Transformer, Convolutional Neural Networks.*

1. Introduction

Hyperspectral images (HSI), in contrast to ordinary RGB photographs, capture a greater variety of spectral bands across the same spatial region. This yields data that is richer and more significant. In remote sensing, HSI categorization is a crucial field of study [1],[2]. Standard Hyperspectral datasets contain data in various spectral bands over the same area that allows us to understand and interpret the geographical aspects efficiently. But there are drawbacks to this abundance of data as well, such the "curse of dimensionality." The model may become less effective as a result of this problem if it takes longer to train.

The methods used for the classification of HSI are divided into two categories: patch-wise and patch-free approaches [1],[3],**Error! Reference source not found.**. In patch-wise approaches, the HSI data is divided into smaller patches, each containing multiple spectral bands. These patches are treated as individual input units, allowing the model to focus on spectral characteristics with neighboring patches. Hence by analyzing spectral information at patch levels, these methods capture detailed spectral signatures and spatial variations within images. On the other hand, patch-free approaches consider the entire HSI image as a single entity without subdividing into patches. These methods prioritize global spatial relationships over spectral details, using techniques like deep convolutional neural networks (CNNs) to extract meaningful spatial features. Patch-wise strategies perform better at capturing both spectral and spatial information at a localized level, which can be advantageous for accurate classification in HSI datasets with complex spectral signatures and spatial variations.

Vision Transformer is inspired from the Transformer architecture, which was originally designed for Natural Language Processing tasks [13]. Unlike traditional patch-free approaches such as CNNs that process entire images, vision transformer model divides the image into fixed size patches and treats each patch as an individual input. These patches are then flattened into a single dimension.

Vision Transformer assigns positional embeddings for each patch, as the spatial arrangement of patches within an image plays a crucial role in understanding its structure. The model uses self-attention mechanism to capture complex relationships between different parts of the image. This is achieved by enabling each patch to interact with every other patch, allowing the model to interpret relationships effectively. Finally, a feed-forward neural network is applied to generate the output.

2. Related Work

In the field of hyperspectral image (HSI) classification, numerous methods have been developed to process the rich spatial and spectral data [2],[19]. These techniques can be broadly categorized into patch-free and patch-wise approaches.

2.1. Patch-Free Approaches

Many patch-free methods consider HSI data in comprehensive manner, they still face severe challenges regarding the computational cost and capturing minute details [1],[3],**Error! Reference source not found.**. In patch-free approaches the model considers the whole image as a single entity. This will definitely lower the computational cost but at the loss of several minute details which are crucial for accurate classification of. To address this, Liu et al. [5] introduced a broad contextual residual network that combines multiscale filters to enable joint examination of local textures and global spatial patterns.

Sidike et al. [6] improved preprocessing through the use of Principal Component Analysis (PCA) for removal of redundant spectral bands prior to extracting texture features with Local Binary Patterns (LBP), an approach very similar to what was adopted in this study. Wang et al. **Error! Reference source not found.** in another study applied a sequential framework that initially compressed hyperspectral data through PCA and subsequently conducted classification by a Random Forest model with a trade-off between efficiency and accuracy. In unison, these studies highlight the significance of PCA as an efficient first step in reducing hyperspectral datasets while maintaining their very critical spectral information [12].

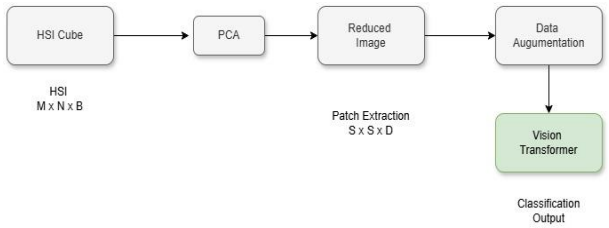

Fig 1: The Architecture of the Vision Transformer Model

2.2. Patch-Wise Approaches

Unlike patch-free techniques, patch-wise techniques segment hyperspectral (HSI) data into small, usually overlapping areas to consider both spatial and spectral attributes at a local scale [3],**Error! Reference source not found.**,**Error! Reference source not found.**,[15]. This approach allows the model to capture detailed spatial variations and minor spectral patterns that lead to more precise classification outcomes. Ahmad et al. [8] proposed a small 3D Convolutional Neural Network (CNN) that handles these overlapping 3D patches in spatial (x, y) and spectral (z) dimensions through 3D convolutional kernels. This architecture enables simultaneous learning of spatial and spectral relationships in the data. Following this concept, Li and Zhang [9] developed FSKNet, a hybrid network that combines 2D-CNN and 3D-CNN modules to strike a balance between computational complexity and feature fulness. Through optimizing the interaction between 2D and 3D convolutions, their model performs effective dimensionality reduction while retaining deep feature extraction. Likewise, Roy et al. [10] proposed the HybridSN model, which uses Principal Component Analysis (PCA) to initially reduce spectral redundancy and then extracts 3D neighbourhood patches from the reduced data cube. These patches are then further processed using a mix of 3D and 2D CNN layers in order to learn hierarchical spectral–spatial representations efficiently [3],[15]. Support vector machines have also been applied for hyperspectral classification [11]. Collectively, these works illustrate that patch-wise architectures are ideally suited to modelling intricate hyperspectral relationships by learning complementary spectral and spatial information at various scales **Error! Reference source not found.**,**Error! Reference source not found.**,[15].

3. Problem Statement

Classifying hyperspectral images is a crucial task in remote sensing, but the inherent complexity of the data presents a fundamental challenge. This richness in data also has negative effects on the model building. A vast and high-dimensional dataset is produced by hyperspectral sensors, which record hundreds of spectral bands for every pixel. This high dimensionality makes it very difficult for sophisticated algorithms to learn distinguishing features without overfitting i.e. learning the general characteristics of a land cover instead of memorizing noise from the training data. This "curse of dimensionality" spikes up the computational costs. On the other hand, conventional methods that require less computing power are also ineffective. In order to prevail, they frequently give spatial information the area's shape as priority over rich spectral details. This oversimplification leads to poor classification, as the unique spectral signature is often the only

way to differentiate between similar land cover classes. Therefore, the core challenge in HSI lies in finding an optimal balance between a method that is computationally feasible and also sophisticated enough to accurately process both spatial and spectral information for the effective classification of diverse land cover.

4. Proposed Methodolgy

In this work, the proposed Vision Transformer approach first preprocess hyper spectral image (HSI) data using Principal Component Analysis (PCA) and then apply a Vision Transformer model for hyperspectral image classification [16]. This approach involves dimensionality reduction of spectral bands followed by patch-based processing using Vision Transformer as shown in **Error! Reference source not found.**.

4.1. Data Normalization and Preprocessing

Before feature extraction, the raw hyperspectral data need to be normalized for the sake of stable convergence of neural network training. The min-max normalization method as shown in equation 1 is used to normalize all spectral values into a fixed range from 0 to 1:

$$I_{norm}(m,n,b) = \frac{I(m,n,b) - I_{min}}{I_{max} - I_{min}} \#(1)$$

Where I_{min} and I_{max} are the minimum and maximum intensity values of all the spectral bands, respectively. Normalization in this step avoids numerical instability in gradient descent optimization and forces all the spectral bands to have equal contribution in the following feature learning process.

4.2. Spectral Dimensionality Reduction Using PCA

Hyperspectral images have hundreds of spectral bands with high correlation and redundancy among many of them [6],[12]. To counter the curse of dimensionality while maintaining discriminative information, Principal Component Analysis is utilized as a linear transformation method.

The normalized hyperspectral cube Inorm is then reshaped into a 2D matrix$\left(\mathrm{X} \in \mathrm{R}^{(\mathrm{M}\times\mathrm{N})\times\mathrm{B}}\right)$, where each row represents the spectral signature of a spatial pixel. PCA then computes the covariance matrix and its eigenvectors as expressed in Equation 2 to identify the principal components which represent maximum variance:

$$\mathrm{C} = \frac{1}{\mathrm{M}\times\mathrm{N}-1}\mathrm{X}^{\mathrm{T}}\mathrm{X}\#(2)$$

The transformation matrix $V_{reduce} \in R^{B\times D}$ is formed by taking the top D eigenvectors of the largest eigenvalues. In this case, $D = 3$ is taken to form a pseudo-RGB representation, which approximates real color images while preserving the most important spectral features. The reduced data is extracted as expressed in Equation 3:

$$X_{pca} = X \cdot V_{reduce}\#(3)$$

which is subsequently reshaped back to spatial dimensions $X_{pca} \in R^{M\times N\times D}$. The three principal components typically capture over 95% of the cumulative variance in the original spectral data, demonstrating effective information compression.

4.3. Spatial Patch Extraction

For integrating spatial context while being computationally efficient, the PCA-transformation of the hyperspectral image is divided into overlapping spatial patches [8]. Every patch $P_{\alpha,\beta}$ of size $S \times S \times D$ is taken about a center pixel at position (α, β) and computes according to Equation 4:

$$P_{\alpha,\beta} = X_{pca}[\alpha - r{:}\,\alpha + r + 1, \beta - r{:}\,\beta + r + 1, :]\#(4)$$

Where $r = \lfloor S/2 \rfloor$ is the patch radius. In this implementation, a patch size of $S = 25$ is utilized with stride of 1 pixel, allowing dense overlapping extraction that captures minute spatial relationships.

Each patch is assigned the class label corresponding to its center pixel $L_{center} = GT(\alpha, \beta)$, where GT denotes the ground truth map. Background pixels (with label 0) are removed from the training set in order to target useful land cover classes.

4.4. Data Augmentation Strategy

Hyperspectral datasets often exhibit severe class imbalance, with certain types of land cover having much fewer labeled examples than others. To address this imbalance and enhance model generalization, geometric data augmentation strategies are enforced on the patches that are extracted [8],[10].

For a given patch P with corresponding label L, the following augmentation operations are performed:

- Horizontal Flip: $P_h = \text{flip}(P, \text{axis} = 1)$
- Vertical Flip: $P_v = \text{flip}(P, \text{axis} = 0)$

These transformations preserve the spectral characteristics while introducing spatial variations, effectively tripling the training dataset size. The augmented dataset $D_{aug} = (P_i, L_i), (P_{h,i}, L_i), (P_{v,i}, L_i)_{i=1}^{N}$ provides improved coverage of spatial orientations without requiring additional labeled data.

4.5. Vision Transformer Architecture

The Vision Transformer model is the central classification engine, utilizing the self-attention mechanism to capture long-range spatial dependencies within hyperspectral patches [13]. Unlike convolutional neural networks that rely on local receptive fields, transformers process the entire patch simultaneously through global attention operations.

1) Patch Embedding

Each input patch $P \in R^{S\times S\times D}$ is first flattened into a 1D vector $p \in R^{S^2\times D}$ and then linearly projected into an embedding space of dimension d_{model} as defined Equation 5:

$$e = W_{embed} \cdot \text{flatten}(P) + b_{embed}\#(5)$$

Where $W_{embed} \in R^{d_{model}\times(S^2\times D)}$ represents the learnable embedding matrix and b_{embed} denotes the bias term. In this work, $d_{model} = 64$ is employed to balance representational capacity with computational efficiency.

2) Positional Encoding and Classification Token

A learnable classification token CLS is added to the sequence of patch embeddings as formulated in Equation 6:

$$z_0 = [e_{cls}; e_1; e_2; \ldots; e_P]\#(6)$$

where $P = (S/p)^2$ is the count of sub-patches formed with patch size $p = 5$. Learnable positional embeddings $E_{pos} \in R^{(P+1)\times d_{model}}$ are used to add spatial relationship encoding as described equation 7:

$$z_0' = z_0 + E_{pos}\#(7)$$

3) Multi-Head Self-Attention Mechanism

The core of the transformer architecture consists of multi-head self-attention (MSA) layers that compute pairwise relationships between all patch positions [13]. For an input sequence (z), three projection matrices generate queries (Q), keys (K), and values (V) as illustrated in Equation 8:

$$Q = zW_Q, \quad K = zW_K, \quad V = zW_V\#(8)$$

The scaled dot-product attention is computed according to Equation 9:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V\#(9)$$

Where $d_k = d_{model}/h$ represents the dimension per attention head, and $\mathrm{h} = 4$ denotes the number of

attention heads. Multi-head attention allows the model to attend to various representation subspaces simultaneously as prescribed in Equation 10:

$$\text{MSA}(z) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \#(10)$$

4) Feed-Forward Network

After every attention layer, a position-wise feed-forward network (FFN) that consists of two linear transformations with GELU activation as defined in Equation 11 is applied:

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2 \#(11)$$

Where $W_1 \in R^{d_{model} \times d_{ff}}$ expands the representation to $d_{ff} = 128$ dimensions, and $W_2 \in R^{d_{ff} \times d_{model}}$ projects back to the original dimension. The FFN introduces non-linearity and increases model capacity.

5) Transformer Encoder Blocks

The entire transformer encoder block is a combination of multi-head self-attention and feed-forward networks with layer normalization and residual connections formulated in Equation 12 and Equation 13 respectively:

$$z'l = \text{MSA}\big(\text{LN}(zl - 1)\big) + z_{l-1} \#(12)$$
$$z_l = \text{FFN}\big(\text{LN}(z_l')\big) + z_l' \#(13)$$

where $\big(\text{LN}(\cdot)\big)$ represents layer normalization and $l \in 1,2,3$ is the index of encoder layers. This work employs 3 transformer blocks to balance model depth with training efficiency.

6) Classification Head

Following processing through all transformer blocks, the final CLS token representation z_L^0 is taken and subjected to a linear classification layer as shown in Equation 14.

$$\hat{y} = \text{softmax}(W_{cls} \cdot z_L^0 + b_{cls}) \#(14)$$

where $W_{cls} \in R^{C \times d_{model}}$ projects the learned representation into $C = 16$ class probabilities corresponding to various land cover types.

4.6. Training Configuration

The model is trained using the cross-entropy loss function, whose formulation is presented in Equation (15):

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log\big(\widehat{y_{i,c}}\big) \#(15)$$

where $y_{i,c}$ represents the one-hot encoded ground truth label. The Adam optimizer with learning rate $\eta = 0.001$ and weight decay $\lambda = 10^{-5}$ is employed for gradient-based optimization. To prevent overfitting, dropout with probability 0.25 is applied to both attention weights and feed-forward layers.

The training dataset is split into 70% for training and 30% for testing using stratified random sampling to maintain class distribution. Mini-batch stochastic gradient descent with batch size 32 is utilized, and the model is trained for 60 epochs with early stopping based on validation performance.

4.7. Model Hyperparameters

Table 1: Parameters Used in Vision Transformer Model

Parameter	Value
PCA Components (D)	3
Patch Size (S)	25 × 25
Sub-patch Size (p)	5 × 5
Embedding Dimension	64
Number of Heads	4
Transformer Depth	3
FFN Dimension	128
Dropout Rate	0.25
Batch Size	32
Learning Rate	0.001
Training Epochs	60

Table presents the key hyperparameters used in the proposed model. The PCA Components (D) represent the number of principal components retained after dimensionality reduction. The Patch Size (S) defines the spatial dimensions of the input patches, while the Sub-patch Size (p) specifies the tokenization patch size utilized by the Transformer. The Embedding Dimension refers to the internal hidden dimension of the Transformer architecture. The Number of Heads indicates how many attention heads are used in the multi-head attention mechanism. The Transformer Depth denotes the number of encoder blocks stacked within the model. The FFN Dimension represents the expansion factor used in the feed-forward network. The Dropout Rate specifies the probability used for regularization to prevent overfitting. The Batch Size refers to the number of samples processed together in one forward and backward pass during training. Finally, the Learning Rate defines the initial step size for the Adam optimizer, and the Training Epochs represent the maximum number of complete passes through the training dataset.

5. Results and Discussion

We randomly split the data into 70% for training and 30% for testing. **Error! Reference source not found.** and **Error! Reference source not found.**, show the confusion matrix of the Vision Transformer model for both the Indian Pines and University of Pavia datasets respectively. Training Loss vs Epochs Curves are shown in **Error! Reference source not found.** and

when performed on Indian Pines and University of Pavia datasets respectively. Training Accuracy vs Epochs Curves are shown in **Error! Reference source not found.** and **Error! Reference source not found.** when performed on Indian Pines and University of Pavia datasets respectively. The Actual vs. predictions were shown in **Error! Reference source not found.** and **Error! Reference source not found.** for Indian Pines and University of Pavia datasets.

The evaluation of classification performance in this study is conducted using three standard metrics: **Overall Accuracy (OA)**, **Average Accuracy (AA)**, and the **Kappa Coefficient (κ)**. $OA = \frac{\sum_{i=1}^{C} x_{ii}}{N} \times 100\%$ where x_{ii} denotes the number of correctly classified samples in class *i*, *C* is the total number of classes, and *N* represents the total number of test samples. In the vision Transformer model, we got overall accuracy **99.58%** and **99.85%** was achieved on the Indian Pines and University of Pavia datasets, respectively. The **Average Accuracy (AA)** metric computes the mean of individual class accuracies, providing a balanced assessment across imbalanced datasets. It is expressed as $AA = \frac{1}{C}\sum_{i=1}^{C} \frac{x_{ii}}{x_{i+}} \times 100\%$ where x_{i+} represents the total number of samples in class *i*. Using this metric, the Vision Transformer model achieved **98.94%** and **99.93%** average accuracy on the Indian Pines and University of Pavia datasets, respectively. Finally, the **Kappa Coefficient (κ)** measures the agreement between predicted and ground truth labels while accounting for chance agreement. It is calculated as $\kappa = \frac{N\sum_{i=1}^{C} x_{ii} - \sum_{i=1}^{C}(x_{i+} \times x_{+i})}{N^2 - \sum_{i=1}^{C}(x_{i+} \times x_{+i})}$. The Vision Transformer achieved Kappa coefficients of **0.9838** and **0.9981** on the Indian Pines and University of Pavia datasets, respectively, demonstrating highly reliable and consistent classification performance.

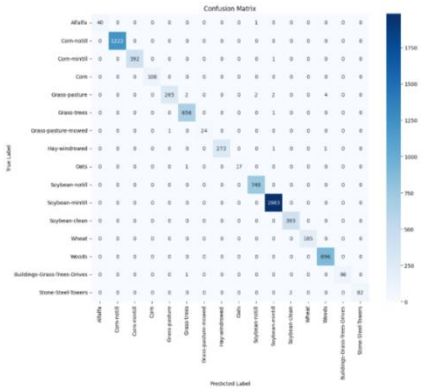

Fig 2: Confusion Matrix for IP Dataset

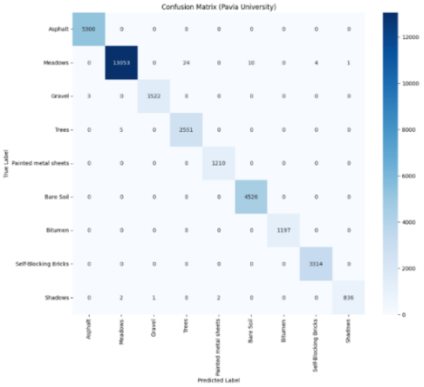

Fig 3: Confusion Matrix for PU dataset

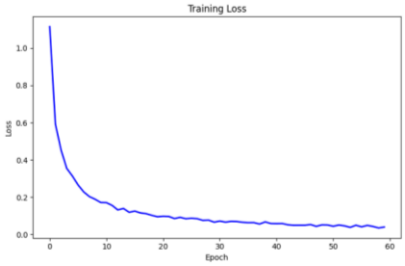

Fig 4: Epochs vs. Training Loss Curve for the Indian Pines Dataset

Fig 5: Epochs vs. Training Accuracy Curve for the Indian Pines Dataset.

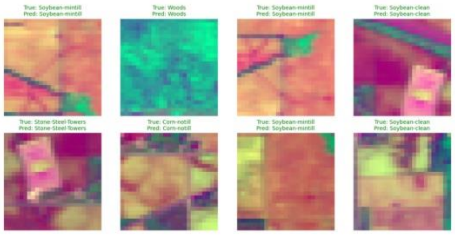

Fig 6: Actual vs. Predicted Output Images for the University of Pavia Dataset.

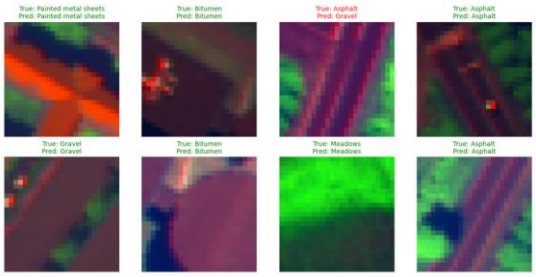

Fig 7: Actual Vs. Predicted Output Images for the Indian Pines Dataset.

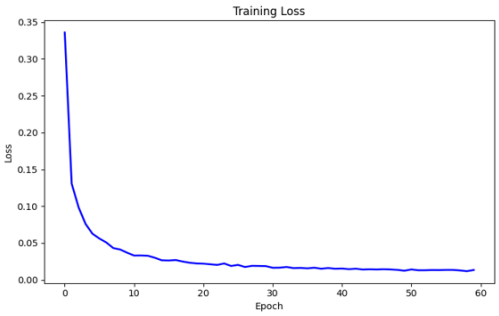

Fig 8: Epochs vs. Training Loss Curve Performed on University Ozf Pavia Dataset

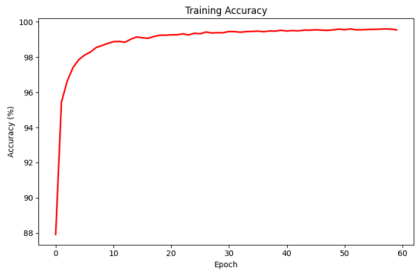

Fig 9: Epochs vs. Training Accuracy curve for the University of Pavia dataset

6. Conclusion

This paper introduced a Principal Component analysis-based Vision Transformer framework for hyperspectral image classification with competitive performance, significantly lower model complexity, and reduced resource usage. Through PCA-based dimensionality reduction and transformer-based spatial-spectral modeling, our method competes with deeper CNN-based models but using only a fraction of the parameters. Experiments on benchmark datasets establish the suitability of the method to remote sensing with efficient and scalable application. Class discrimination can further be improved in future work using non-linear reduction and hybrid models.

References

[1] Li, Shutao, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jon Atli Benediktsson. "Deep learning for hyperspectral image classification: An overview." *IEEE transactions on geoscience and remote sensing* 57, no. 9 (2019): 6690-6709.

[2] Lv, Wenjing, and Xiaofei Wang. "Overview of hyperspectral image classification." *Journal of Sensors* 2020, no. 1 (2020): 4817234.

[3] He, Lin, Jun Li, Chenying Liu, and Shutao Li. "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines." *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 3 (2017): 1579-1597.

[4] Gunda, S. K. G. (2023). The Future of Software Development and the Expanding Role of ML Models. International Journal of Emerging Research in Engineering and Technology, 4(2), 126-129. https://doi.org/10.63282/3050-922X.IJERET-V4I2P113

[5] S. Liu, H. Luo, Y. Tu, Z. He, and J. Li, "Wide contextual residual network with active learning for remote sensing image classification," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 7145–7148.

[6] P. Sidike, C. Chen, V. Asari, Y. Xu, and W. Li, "Classification of hyperspectral image using multiscale spatial texture features," in 2016 *8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing* (WHISPERS). IEEE, 2016, pp. 1–4.

[7] S. K. Gunda, "Software Defect Prediction Using Advanced Ensemble Techniques: A Focus on Boosting and Voting Method," 2024 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 2024, pp. 157-161, https://doi.org/10.1109/ICESIC61777.2024.10846550.

[8] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-d cnn for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.

[9] G. Li and C. Zhang, "Faster hyperspectral image classification based on selective kernel mechanism using deep convolutional networks," *arXiv preprint arXiv:*2202.06458, 2022.

[10] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-d–2-d cnn feature hierarchy for hyperspectral image classi f ication," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, 2019.

[11] Moughal, T. A. "Hyperspectral image classification using support vector machine." In *journal of physics: conference series*, vol. 439, no. 1, p. 012042. IOP Publishing, 2013.

[12] Rodarmel, Craig, and Jie Shan. "Principal component analysis for hyperspectral image classification." *Surveying and Land Information Science* 62, no. 2 (2002): 115-122.

[13] Zhao, Zhuoyi, Xiang Xu, Shutao Li, and Antonio Plaza. "Hyperspectral image classification using groupwise separable convolutional vision transformer network." *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024): 1-17.

[14] Krishna GV, Reddy BD, Vrindaa T. EmoVision: An Intelligent Deep Learning Framework for Emotion Understanding and Mental Wellness Assistance in Human Computer Interaction. 2025 Oct ;6(4):14-20. Available from: https://ijaidsml.org/index.php/ijaidsml/article/view/295

[15] Banerjee, Anasua, Minakhi Rout, and Mainak Bandyopadhyay. "Spectral & Spatial Feature detection on Hyperspectral Images using Deep Neural Networks." In *2021 2nd International Conference on Range Technology (ICORT)*, pp. 1-6. IEEE, 2021.

[16] Banerjee, Anasua, and Debajyoty Banik. "Resnet-2D-ConvLSTM: A means to extract features from hyperspectral image." In *International conference on neural information processing*, pp. 365-376. Singapore: Springer Nature Singapore, 2022.

[17] Banerjee, Anasua, Satyajit Swain, Minakhi Rout, and Mainak Bandyopadhyay. "Composite spectral spatial pixel CNN for land-use hyperspectral image classification with hybrid activation function." *Multimedia Tools and Applications* 84, no. 12 (2025): 10527-10550.

[18] Hamida, Amina Ben, Alexandre Benoit, Patrick Lambert, and Chokri Ben Amar. "3-D deep learning approach for remote sensing image classification." *IEEE Transactions on geoscience and remote sensing* 56, no. 8 (2018): 4420-4434.

[19] Datta, Debaleena, Pradeep Kumar Mallick, Akash Kumar Bhoi, Muhammad Fazal Ijaz, Jana Shafi, and Jaeyoung Choi. "Hyperspectral image classification: Potentials, challenges, and future

directions." *Computational intelligence and neuroscience* 2022, no. 1 (2022): 3854635.

[20] Sai Krishna Gunda; An exploration of adaptive ensemble approaches in software fault detection: Balancing accuracy and robustness. *AIP Conf. Proc.* 7 January 2026; 3345 (1): 020211. https://doi.org/10.1063/5.0298093

[21] Berroukham, Abdelhafid, Khalid Housni, and Mohammed Lahraichi. "Vision transformers: a review of architecture, applications, and future directions." In *2023 7th IEEE congress on information science and Technology (CiSt)*, pp. 205-210. IEEE, 2023.