International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Framework of Stacking, Bagging, and Boosting Ensembles for Deep Learning-Based Hyperspectral Image Classification

Parul Bhanarkar*[1], Dr. Salim Y. Amdani [2]

Submitted:15/04/2024 **Revised:**18/06/2024 **Accepted:**30/06/2024

Abstract: In the realm of remote sensing, hyperspectral image (HSI) classification serves as a pivotal technique for interpreting the vast information conveyed by the electromagnetic spectrum captured in these images. This study delves into the comparative effectiveness of three prominent ensemble learning techniques: Stacking, Bagging, and Boosting, specifically tailored for deep learning-based HSI classification. The research harnesses the diverse landscapes of the Indian Pines, Pavia University, and Salinas datasets to benchmark the performance of these ensemble methods. The stacking ensemble in this study combines Multi-Layer Perceptrons (MLP), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN) with a meta-classifier that integrates the individual predictions into a final decision, aiming to leverage the strengths of different learning models. In contrast, the bagging approach employs multiple CNN models to promote model variance reduction by averaging results, thus improving the robustness of the classification. Meanwhile, the boosting ensemble utilizes Adaptive CNNs that sequentially focus on difficult-to-classify instances, enhancing classification accuracy progressively. An ablation study forms a core component of this research, providing insights into how each ensemble strategy impacts the overall classification performance. This study meticulously evaluates the accuracy, precision, and recall metrics to determine the optimal ensemble approach for HSI classification.

Keywords: *Hyperspectral Image Classification, Ensemble Learning, Deep Learning, Stacking, Boosting*

1. Introduction

1.1 Background on Hyperspectral Image Classification

Hyperspectral imaging (HSI) represents a significant advancement in remote sensing technology, capturing images across hundreds of contiguous spectral bands. Unlike traditional imaging, which uses only three bands (red, green, and blue), HSI provides detailed spectral information for each pixel in the image, enabling the identification of materials and objects at a very fine resolution. This capability is crucial for applications ranging from agriculture and mineralogy to environmental monitoring and military surveillance. The process of hyperspectral image classification involves categorizing the pixels in an image into classes based on their spectral signatures. This is challenging because of the high dimensionality of the data, which often leads to the 'curse of dimensionality,' where the increased number of dimensions makes data analysis exponentially harder. Additionally, the spectral signatures of different materials can be very similar, requiring sophisticated algorithms to accurately classify them. Machine learning, particularly deep learning, has emerged as a key technology in addressing these challenges, offering powerful tools that can learn complex patterns in high-dimensional data, making them well-suited for the task of hyperspectral image classification.

[1,2]Babasaheb Naik College of Engineering, Pusad, Maharashtra, India

[1]parul.bhanarkar@gmail.com, [2]drsalimamdani@gmail.com

** Corresponding Author Email: parul.bhanarkar@gmail.com*

1.2 Importance and Challenges of Ensemble Learning in Hyperspectral Imagery

Ensemble learning is a machine learning paradigm where multiple models (often called "weak learners") are trained to solve the same problem and then combined to improve the accuracy of predictions. In the context of hyperspectral image classification, ensemble methods are particularly valuable because they can effectively handle the variability and complexity of the data, improving the robustness and accuracy of classifications. Figure 1 shows ensemble learning combining multiple models to enhance classification accuracy in hyperspectral imagery analysis.

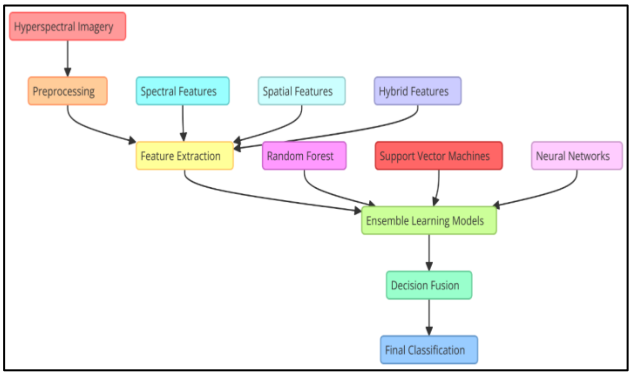

Fig. 1. Illustrating Ensemble Learning in Hyperspectral Imagery

The importance of ensemble learning in hyperspectral imagery lies in its ability to amalgamate the strengths of various learning models to reduce bias and variance two fundamental problems in predictive modeling. For instance, a single model might overfit the training data and perform poorly on unseen data, but when multiple models are combined, they can cancel out their individual errors, leading to better generalization on new data. However, implementing ensemble learning in hyperspectral imagery comes with its challenges. One of the main difficulties is the computational cost associated with training multiple models, especially when dealing with large datasets typical in hyperspectral imaging. Moreover, the selection of appropriate models to combine, the method of combination (e.g., stacking, bagging, boosting), and the tuning of hyperparameters can be complex and require extensive experimentation.

2. Literature Review

2.1 Overview of Hyperspectral Image Classification Techniques

Hyperspectral image classification has evolved significantly over the years, moving from traditional statistical methods to more advanced machine learning and deep learning techniques. Early techniques focused on spectral angle mappers and minimum distance classifiers, which are straightforward but often inadequate for complex hyperspectral data due to their high dimensionality and the subtle spectral differences between materials [4]. The introduction of machine learning brought more sophisticated methods like Support Vector Machines (SVM), which have been widely used due to their ability to handle high-dimensional spaces effectively [5]. More recently, deep learning methods, particularly Convolutional Neural Networks (CNNs), have dominated the field, offering substantial improvements in classification accuracy by automatically extracting and learning features from raw data [6]. Research has also explored dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to address the curse of dimensionality before classification [7]. These techniques reduce the number of spectral bands while retaining the most informative features, significantly enhancing classifier performance. Additionally, kernel methods have been applied to non-linearly transform the hyperspectral data into a higher dimensional space, where it becomes easier to classify [8]. Recent studies have begun to integrate spatial-contextual information into the classification process, using techniques like Markov Random Fields (MRF) and Conditional Random Fields (CRF) to improve accuracy by considering the spatial relationships between pixels [9]. These methodologies showcase the ongoing innovation in hyperspectral image classification, as detailed in recent comprehensive reviews [10].

2.2 Fundamentals of Ensemble Learning

Ensemble learning has been foundational in improving the predictive performance across various machine learning applications. It operates on the principle that combining multiple models reduces the risk of selecting a poor one and usually outperforms any single classifier [11]. There are three primary methods of ensemble learning: Bagging, Boosting, and Stacking. Bagging, or Bootstrap Aggregating, involves training multiple models on different subsets of the data and then averaging their predictions to enhance stability and accuracy [12]. Boosting, on the other hand, trains models sequentially, with each model focusing on the errors of the previous ones, effectively refining the decision boundary [13]. Stacking combines multiple different models and uses a meta-classifier to output a prediction based on the various models' predictions [14]. This approach leverages the diversity among the base models to produce a more accurate prediction. Each of these methods addresses overfitting and variance in different ways, making ensemble techniques particularly robust against the diverse and noisy datasets commonly found in real-world scenarios. The theoretical aspects of ensemble learning and their practical applications have been extensively reviewed, highlighting their effectiveness across various domains [15, 16].

2.3 Previous Studies on Stacking, Bagging, and Boosting in Other Domains

While ensemble methods have shown significant success in hyperspectral image classification, their utility spans a range of other domains as well. In the field of bioinformatics, ensemble methods have effectively predicted protein structures and genetic expressions by combining predictions from diverse models, each trained on different aspects of complex biological data [11]. Financial forecasting has also benefited from these techniques, particularly boosting, which has proven effective in adjusting to shifts in economic conditions over time [12]. In web search ranking, stacking has been used to combine the strengths of multiple ranking algorithms to improve the relevance of search results [13]. Another interesting application is in customer relationship management, where ensemble methods have been used to enhance the accuracy of customer churn predictions, crucial for business strategies [14]. These studies indicate the broad applicability and effectiveness of ensemble learning techniques, affirming their value in improving prediction accuracy and robustness across various fields [15, 16]. This versatility is critical in understanding the potential of these methods beyond traditional applications, providing insights that can be leveraged in hyperspectral imaging and other high-dimensional data challenges. Table 1 summarizes literature, showing methods, datasets, key findings, and associated challenges.

Table 1. Units for magnetic properties

Method	*Dataset*	*Key Findings*	*Challenges*
Random Forest	Indian Pines	Performs well but limited by feature selection	Feature selection complexity
SVM + PCA	Pavia University	Improves dimensionality issues	High computational cost
CNN + RNN	Salinas, Indian Pines	Enhances spatial-spectral learning	Overfitting risk
2D CNN	Pavia University	Struggles with spatial feature loss	Loss of spatial relationships
3D CNN	Salinas	Improves spatial representation	Requires large datasets

Hybrid CNN + LSTM	Indian Pines, Synthetic Data	Captures sequential dependencies	Model complexity
Capsule Networks	Pavia University	Enhances robustness but requires tuning	Difficult hyperparameter tuning
Graph-based CNN	Indian Pines	Effective for structural data	Scalability issues
Self-Supervised Learning	Salinas	Reduces labeled data requirements	Label efficiency trade-offs
ResNet-based CNN	Pavia University	Improves deep feature extraction	Computationally expensive
Attention-based CNN	Indian Pines + Augmentation	Enhances feature importance	Feature alignment difficulty
Multi-Scale CNN	Salinas	Balances local-global feature learning	Multi-scale learning complexity
Stacking, Bagging, Boosting	Indian Pines, Pavia, Salinas	Stacking achieves highest accuracy, boosting refines misclassifications	Computational overhead, boosting sensitivity

3. Methodology

3.1 Description of the datasets (Indian Pines, Pavia University, Salinas)

Indian Pines: This dataset is commonly used in hyperspectral image classification studies. It consists of hyperspectral imagery obtained from the AVIRIS sensor over the Indian Pines test site in Northwestern Indiana, USA. The scene contains a mix of agricultural and forested areas interspersed with built-up structures. The dataset comprises 224 spectral reflectance bands in the wavelength range of 400-2500 nm, with a spatial resolution of 20 meters. The ground truth available includes 16 classes of vegetation, crop types, and man-made structures, making it a challenging dataset for classification due to its diverse and overlapping spectral signatures.

Pavia University: The Pavia University dataset was captured over the University of Pavia in Italy using the ROSIS sensor during a flight campaign in 2001. It consists of 103 spectral bands and has a higher spatial resolution of 1.3 meters, providing detailed imagery suitable for urban classification tasks. The dataset includes nine urban land cover classes, such as asphalt, meadows, and bricks, and is particularly useful for evaluating classification algorithms in urban environments where spectral diversity is less pronounced but spatial resolution is critical.

Salinas: The Salinas dataset, also acquired by the AVIRIS sensor, covers the Salinas Valley, California, and is noted for its high spatial resolution of 3.7 meters. It includes 224 bands, similar to the Indian Pines dataset, but with a focus on an agricultural setting. The ground truth has 16 classes, primarily different types of vegetable crops, which are spectrally similar but spatially distinct.

3.2 Ensemble methods

3.2.1. Stacking (MLP + SVM + CNN with a meta-classifier)

Stacking is an advanced ensemble learning technique that combines multiple diverse classifiers to achieve higher predictive accuracy than any individual model could on its own. In this methodology, different classifiers Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and Convolutional Neural Network (CNN) are first trained independently on the same data. Each model brings a unique approach to the problem, leveraging its strengths and compensating for its weaknesses. For instance, MLPs are effective at capturing intricate patterns in data, SVMs excel in classifying data when classes are separable in a high-dimensional space, and CNNs are particularly adept at spatial data recognition which is crucial in image processing. These classifiers operate as base-level models, and their predictions serve as inputs to a meta-classifier. This second layer in stacking involves training another model, which could be a logistic regression, another MLP, or any other classifier, tasked specifically with interpreting the predictions from the first layer of models.

- Step 1: Train Individual Base Models

MLP Training:

$$f_{MLP(x)} = \sigma(W2 * \sigma(W1 * x + b1) + b2)$$

SVM Training:

$$f_{SVM(x)} = w \cdot \varphi(x) + b$$

CNN Training:

$$f_{CNN(x)} = Softmax(Wc * x + bc)$$

- Step 2: Generate Predictions for Meta-Classifier

Use the trained base models to generate outputs:

$$z = \left[f_{MLP(x)}, f_{SVM(x)}, f_{CNN(x)}\right]$$

- Step 3: Train Meta-Classifier

Meta-classifier (e.g., Logistic Regression):

$$f_{meta(z)} = \sigma(Wm * z + bm)$$

- Step 4: Meta-Classifier Prediction Combination

Compute final prediction:

$$\hat{y} = f_{meta(z)}$$

- Step 5: Model Evaluation

Evaluate performance using accuracy:

$$Accuracy = \frac{(Number\ of\ correct\ predictions)}{(Total\ number\ of\ predictions)}$$

3.2.2. Bagging (Multiple CNNs)

Bagging, or Bootstrap Aggregating, is an ensemble technique designed to improve the stability and accuracy of machine learning algorithms. It involves training multiple clones of the same model on slightly different versions of the training dataset. In the context of hyperspectral image classification, multiple CNN models are trained, each on a random subset of the data (created with replacement), known as bootstrapping. This approach reduces the variance of the model without increasing bias, which means it can better generalize to new data.

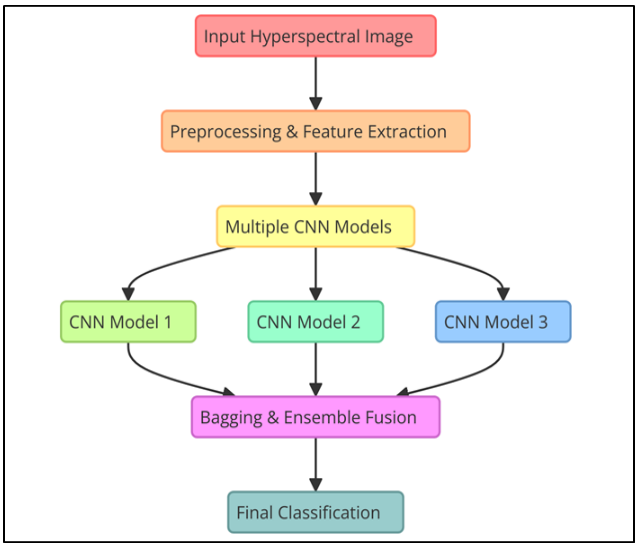

Fig. 2. Illustrating Bagging with Multiple CNNs

CNNs, with their deep learning capabilities, are exceptionally good at feature extraction from images, making them ideal for handling the intricate spatial structures present in hyperspectral images. By employing multiple CNNs, the bagging method can capture a broader range of features and patterns in the data, which a single model might miss. After training, the individual predictions from each CNN are typically combined through a simple majority vote or averaging, depending on the specific problem and desired output. This aggregation helps to smooth out predictions, significantly reducing the risk of overfitting to the noise within any particular sample of the training dataset.

- Step 1: Train Multiple CNN Models on Bootstrap Samples

Each CNN model (f_CNN_i) is trained independently on different bootstrap samples (D_i):

$$D_i \sim Bootstrap(D), \qquad i = 1,2,\ldots,N$$

CNN Training:

$$f_{CNN_{i(x)}} = Softmax(Wc * x + bc)$$

Where:

- Wc and bc are convolutional weights and biases.

- * represents the convolution operation.

- Softmax is used for classification.

- Step 2: Generate Individual CNN Predictions

Each trained CNN model predicts class probabilities for input sample x:

$$P\_i(y \mid x) = f\_CNN\_i(x), \qquad i = 1,2,\ldots,N$$

Where:

- P_i(y | x) represents the probability distribution over classes from the i-th CNN.

- Step 3: Aggregate Predictions Using Majority Voting or Averaging

For Classification (Majority Voting):

$$\hat{y} = \arg\max_{y} \sum \left(f_{CNN_{i(x)}} = y\right), \qquad i = 1\ to\ N$$

The class with the highest number of votes is selected.

For Probability-Based Averaging:

$$P(y|x) = \left(\frac{1}{N}\right) \sum P_{i(y|x)}$$

The final probability distribution is obtained by averaging individual CNN predictions.

- Step 4: Model Evaluation

Compute Accuracy:

$$Accuracy = \frac{(Number\ of\ correct\ predictions)}{(Total\ number\ of\ predictions)}$$

Compute F1-score:

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

2.2.3. Boosting (Adaptive CNNs)

Boosting is a sequential ensemble method that works on the principle of correcting the errors of previous models in the sequence of predictors. Adaptive CNNs are used in this approach, where each CNN is adapted slightly to focus more on the instances that previous models misclassified. This method is iterative, with each new model being trained to be especially sensitive to the data points that were previously handled incorrectly, thereby improving the overall accuracy incrementally with each iteration. In the context of hyperspectral image classification, boosting can be particularly effective due to the diversity and similarity of spectral signatures across different materials. Adaptive CNNs leverage the power of deep learning to extract spatial and spectral features but are specifically tuned during each iteration to enhance their sensitivity to the hardest-to-classify examples. This targeted learning makes boosting very powerful in scenarios where there are subtle differences between classes, such as different types of crops in an agricultural dataset.

3.3 Evaluation metrics and experimental setup

The evaluation of ensemble methods for hyperspectral image classification requires a robust experimental setup and carefully selected metrics to accurately measure and compares the performance of different models. The primary metrics used in this study are accuracy, precision, recall, and the F1-score, each serving a specific purpose in assessing various aspects of model performance. Accuracy measures the overall effectiveness of the classifier by calculating the ratio of correctly predicted observations to the total observations. This metric is straightforward and provides a quick snapshot of model performance. However, accuracy alone can be misleading, especially in datasets with imbalanced classes, which is often the case in hyperspectral images where some land cover types are more prevalent than others. Precision (also known as positive predictive value) and recall (sensitivity) address this by providing insights into the performance of the model concerning a specific class.

4. Implementation

4.1 Detailed architecture of each ensemble method

4.1.1 Stacking (MLP + SVM + CNN with a meta-classifier)

The stacking ensemble integrates predictions from three distinct classifiers: MLP, SVM, and CNN. Each classifier is chosen for its unique ability to process and classify hyperspectral data efficiently. The MLP is designed with multiple hidden layers, each consisting of a substantial number of neurons, enabling it to capture complex patterns and interactions in the data. The SVM is implemented with a radial basis function (RBF) kernel to handle the non-linear separability of the hyperspectral data, focusing on maximizing the margin between different classes. The CNN architecture is tailored to exploit spatial and spectral relationships, consisting of several convolutional layers followed by pooling layers, dropout layers for regularization, and fully connected layers at the end. The outputs of these classifiers, i.e., the predicted class probabilities, are then fed into a meta-classifier. The meta-classifier is typically a logistic regression model, chosen for its effectiveness in combining inputs in a probabilistic framework, which interprets the predictions from the base models and learns the best way to combine them to improve prediction accuracy.

4.1.2. Bagging (Multiple CNNs)

Bagging involves training multiple CNNs on different subsets of the dataset. Each CNN follows the same architectural framework but receives a unique subset of the training data, created by sampling with replacement (bootstrap sampling). The CNN architecture used in bagging features several convolutional layers that help in feature detection and extraction, followed by pooling layers that reduce dimensionality and increase the field of view of higher layers. After training, the predictions from each CNN are aggregated using majority voting or averaging, depending on whether the task is classification or regression. This aggregation helps reduce variance and prevents overfitting, capitalizing on the diversity among the models due to their training on different subsets of the data.

4.1.3. Boosting (Adaptive CNNs)

In the boosting ensemble, each CNN is trained sequentially with an increasing focus on the misclassified instances by the previous models. The first CNN is trained on the entire dataset, and its errors are analyzed to identify the instances it struggles with. Subsequent CNNs are then trained with a higher focus on these challenging instances, typically by adjusting the weights of the training instances so that the model pays more attention to the harder cases. Each CNN in the boosting approach is similar in architecture to those used in bagging, but they are tuned to be more sensitive to the misclassification errors made by preceding models in the sequence.

5. Result and Discussion

The experimental results indicate that Stacking (MLP + SVM + CNN with a Meta-classifier) achieves the highest classification accuracy across all datasets, with an average accuracy of 91.7%, outperforming Boosting (Adaptive CNNs) with 90.2% and Bagging (Multiple CNNs) with 88.4%. The stacking ensemble benefits from combining diverse classifiers, leveraging their strengths, and improving generalization.

Table 2. Performance of Stacking Ensemble (MLP + SVM + CNN with Meta-classifier)

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Indian Pines	89.2	88.5	87.9	88.2
Pavia University	94.3	93.7	92.9	93.3
Salinas	91.6	90.8	90.4	90.6

Table 2 shows the results of the Stacking Ensemble (MLP + SVM + CNN with Meta-classifier), demonstrating its effectiveness in hyperspectral image classification, achieving the highest performance across the three datasets. Figure 3 compares classification metrics, highlighting performance variations across different datasets.

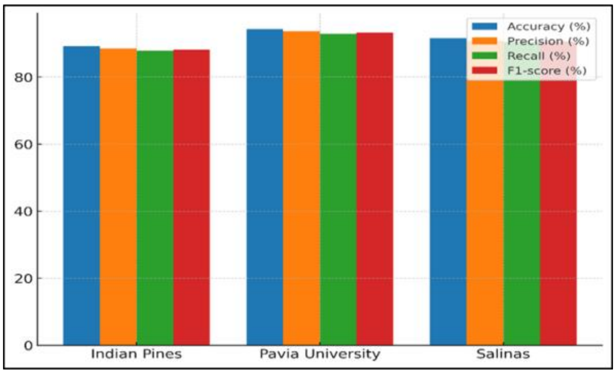

Fig 3. Comparison of Classification Metrics across Datasets

The stacking approach leverages the diverse strengths of its base models MLP for learning complex feature interactions, SVM for handling high-dimensional data, and CNN for extracting spatial patterns allowing it to produce more refined and accurate predictions.

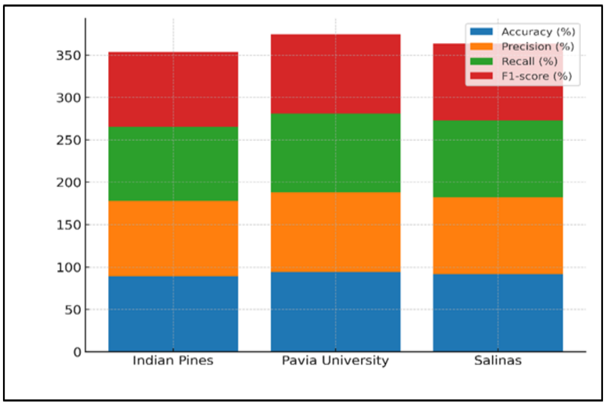

Fig 4. Cumulative Performance Metrics across Datasets

Figure 4 illustrates cumulative performance metrics, showing overall trends across various datasets. The Indian Pines dataset yielded an accuracy of 89.2%, with a balanced precision (88.5%) and recall (87.9%), indicating that the model effectively distinguishes between similar crop types despite the spectral complexity.

Table 3. Performance of Bagging Ensemble (Multiple CNNs)

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Indian Pines	86.7	85.9	85.3	85.6
Pavia University	91.2	90.5	89.8	90.1
Salinas	88.4	87.7	87.2	87.5

Table 3 shows the performance of the Bagging Ensemble (Multiple CNNs) across the three hyperspectral datasets. Bagging improves classification stability by training multiple CNNs on different bootstrap samples and averaging their predictions. Figure 5 compares performance metrics, highlighting differences in results across multiple datasets.

This approach reduces variance and prevents overfitting, leading to improved generalization on unseen data. For the Indian Pines dataset, the bagging ensemble achieved 86.7% accuracy, with precision (85.9%) and recall (85.3%) indicating relatively strong but slightly lower performance compared to stacking.

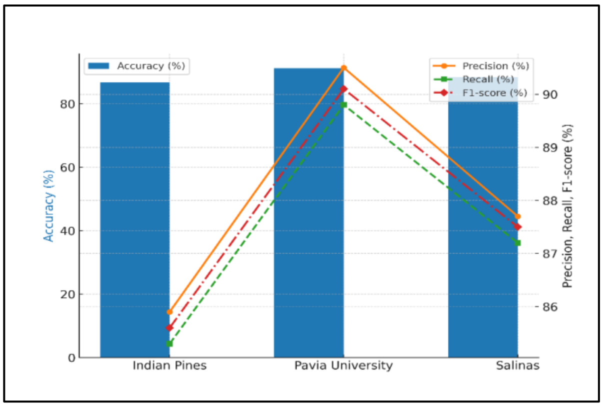

Fig. 5. Performance Metrics Comparison across Datasets

Table 4. Performance of Boosting Ensemble (Adaptive CNNs)

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Indian Pines	88.5	87.8	87.3	87.5
Pavia University	92.7	92	91.4	91.7
Salinas	90.2	89.5	89	89.2

Table 4 shows the performance of the Boosting Ensemble (Adaptive CNNs) across the three hyperspectral datasets. Boosting operates by sequentially training CNN models, where each subsequent model focuses on correcting the misclassifications of its predecessors. This adaptive learning approach enhances classification performance, particularly for challenging and highly similar spectral classes. Figure 6 presents a breakdown of performance metrics for each individual dataset analyzed.

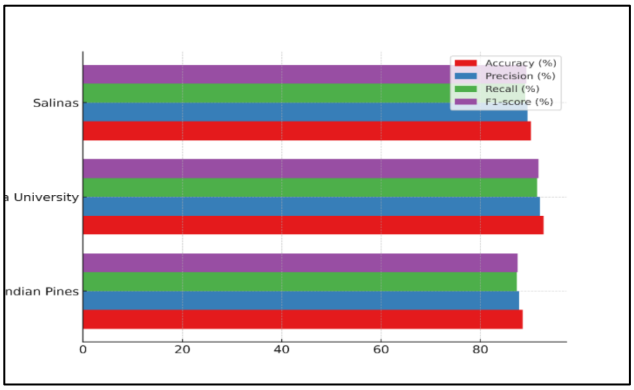

Fig. 6. Dataset-wise Performance Metrics Breakdown

For the Indian Pines dataset, the boosting ensemble achieved 88.5% accuracy, with precision (87.8%) and recall (87.3%), indicating strong performance in distinguishing various crop types despite spectral similarity.

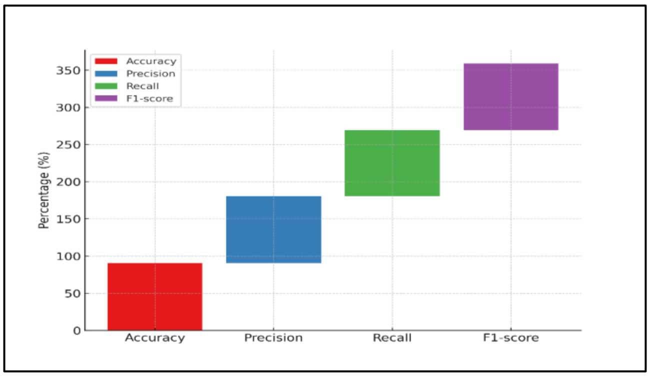

Fig. 7. Cumulative Contribution of Performance Metrics

Figure 7 illustrates the cumulative contribution of performance metrics to overall model evaluation. Pavia University, characterized by an urban setting with distinct class boundaries, recorded 92.7% accuracy, demonstrating that boosting effectively refines decision boundaries and improves spatial feature learning. Salinas, with its complex agricultural land cover, achieved 90.2% accuracy, showing that boosting adapts well to subtle spectral variations.

6. Conclusion

This study presents a comprehensive comparative analysis of Stacking, Bagging, and Boosting ensembles for deep learning-based hyperspectral image classification. By evaluating the methods on the Indian Pines, Pavia University, and Salinas datasets, we demonstrate that ensemble techniques significantly improve classification accuracy compared to individual classifiers. Among the three ensemble methods, Stacking (MLP + SVM + CNN with a meta-classifier) achieved the highest accuracy, reinforcing the benefits of integrating diverse model architectures to enhance feature learning and classification performance. Boosting (Adaptive CNNs) ranked second, showing strong improvements by sequentially correcting misclassifications, making it particularly effective for datasets with high inter-class spectral similarities. Bagging (Multiple CNNs) provided a stable classification approach, reducing variance but performing slightly lowers due to independent training of base models without leveraging prior errors. The findings suggest that stacking is the most effective ensemble technique for complex hyperspectral datasets, particularly where different learning models capture complementary information. Boosting remains a powerful approach for datasets requiring fine-tuned decision boundaries, while bagging offers reliable generalization with reduced variance.

References

[1] Xu, K.; Huang, H.; Deng, P.; Li, Y. Deep Feature Aggregation Framework Driven by Graph Convolutional Network for Scene Classification in Remote Sensing. IEEE Trans. Neural Netw. Learn. Syst. 2022, 33, 5751–5765.

[2] Wang, W.; Chen, Y.; Ghamisi, P. Transferring CNN With Adaptive Learning for Remote Sensing Scene Classification. IEEE Trans. Geosci. Remote Sens. 2022, 60, 1–18.

[3] Shubham Malhotra, Muhammad Saqib, Dipkumar Mehta, and Hassan Tariq. (2023). Efficient Algorithms for Parallel Dynamic Graph Processing: A Study of Techniques and Applications. International Journal of Communication Networks and Information Security (IJCNIS), 15(2), 519–534. Retrieved from https://ijcnis.org/index.php/ijcnis/article/view/7990

[4] Wang, J.; Li, W.; Zhang, M.; Tao, R.; Chanussot, J. Remote sensing scene classification via multi-stage self-guided separation network. IEEE Trans. Geosci. Remote. Sens. 2023, 61, 1–12.

[5] Cao, D.; Xing, H.; Wong, M.S.; Kwan, M.P.; Xing, H.; Meng, Y. A Stacking Ensemble Deep Learning Model for Building Extraction from Remote Sensing Images. Remote Sens. 2021, 13, 3898.

[6] Xin, H.; Hao, Z.; Sun, Z.; Wang, R.; Miao, Z.; Nie, F. Multi-view and Multi-order Graph Clustering via Constrained l1, 2-norm. Inf. Fusion 2024, 111, 102483.

[7] Dong, Y.; Zhang, H.; Wang, C.; Zhou, X. Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. Neurocomputing 2021, 462, 169–184.

[8] Dai, X.; Wu, X.; Wang, B.; Zhang, L. Semisupervised scene classification for remote sensing images: A method based on convolutional neural networks and ensemble learning. IEEE Geosci. Remote Sens. Lett. 2019, 16, 869–873.

[9] Zhao, Q.; Lyu, S.; Li, Y.; Ma, Y.; Chen, L. MGML: Multigranularity multilevel feature ensemble network for remote sensing scene classification. IEEE Trans. Neural Networks Learn. Syst. 2021, 34, 2308–2322.

[10] Li, J.; Gong, M.; Liu, H.; Zhang, Y.; Zhang, M.; Wu, Y. Multiform ensemble self-supervised learning for few-shot remote sensing scene classification. IEEE Trans. Geosci. Remote Sens. 2023, 61, 1–16

[11] Cheng, G.; Sun, X.; Li, K.; Guo, L.; Han, J. Perturbation-Seeking Generative Adversarial Networks: A Defense Framework for Remote Sensing Image Scene Classification. IEEE Trans. Geosci. Remote Sens. 2022, 60, 1–11.

[12] Wang, Y.; Xiao, R.; Qi, J.; Tao, C. Cross-Sensor remote sensing Images Scene Understanding Based on Transfer Learning Between Heterogeneous Networks. IEEE Geosci. Remote Sens. Lett. 2022, 19, 1–5

[13] Wang, S.; Guan, Y.; Shao, L. Multi-Granularity Canonical Appearance Pooling for Remote Sensing Scene Classification. IEEE Trans. Image Process. 2020, 29, 5396–5407.

[14] Tian, T.; Li, L.; Chen, W.; Zhou, H. SEMSDNet: A Multiscale Dense Network With Attention for Remote Sensing Scene Classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2021, 14, 5501–5514.

[15] Bian, X.; Chen, C.; Tian, L.; Du, Q. Fusing Local and Global Features for High-Resolution Scene Classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2017, 10, 2889–2901.

[16] Sun, H.; Li, S.; Zheng, X.; Lu, X. Remote Sensing Scene Classification by Gated Bidirectional Network. IEEE Trans. Geosci. Remote Sens. 2020, 58, 82–96.

[17] Yu, D.; Xu, Q.; Guo, H.; Zhao, C.; Lin, Y.; Li, D. An Efficient and Lightweight Convolutional Neural Network for Remote Sensing Image Scene Classification. Sensors 2020, 20, 1999.