International Interdisciplinary Virtual Conference on
'Recent Advancements in Computer Science, Management and Information Technology'
International Journal of Scientific Research in Computer Science,
Engineering and Information Technology | ISSN : 2456-3307 (www.ijsrcseit.com)

Enhancing Hyperspectral Image Classification with Deep Bagging Ensembles

[1]Parul Bhanarkar, [2]Dr. Salim Y. Amdani
[1,2]Babasaheb Naik College of Engineering, Pusad, Maharashtra, India.
[1]parul.bhanarkar@gmail.com, [2]drsalimamdani@gmail.com

ABSTRACT

Hyperspectral imaging, with its rich spectral information across numerous contiguous spectral bands, provides unprecedented opportunities for precise pixel classification. However, the complexity and high dimensionality of hyperspectral data often pose significant challenges, such as Hughes phenomenon and overfitting, particularly when using deep learning models like Convolutional Neural Networks (CNNs). This research introduces a novel approach to hyperspectral image classification that leverages the robustness of bagging (bootstrap aggregation) integrated with CNNs to enhance classification accuracy and model stability. By implementing a deep bagging ensemble, where multiple CNN models are trained independently on random subsets of the training data and then aggregated, the proposed method aims to reduce model variance without substantially increasing bias. Each CNN acts as a base classifier, operating on bootstrap samples derived from the original dataset, thereby introducing diversity in the model predictions. The ensemble's final decision is made through majority voting or averaging, which improves generalization over unseen data by mitigating the risk of overfitting to noisy or unrepresentative training samples. Comparative analysis with traditional single-model approaches and other ensemble techniques confirms the efficacy of the deep bagging ensemble in achieving superior classification performance. The method not only addresses the spectral-spatial classification challenges in hyperspectral datasets but also shows significant improvement in stability and accuracy across various challenging datasets. This study's outcomes suggest that deep learning ensembles, particularly those utilizing bagging with CNNs, can effectively tackle the intricacies of hyperspectral image classification, making them suitable for advanced remote sensing applications.

Keywords: Hyperspectral Imaging, CNN, Bagging, Ensemble Learning, Pixel Classification, Model Stability

1. Introduction

Hyperspectral imaging, or HSI, is a device that can gather and handle data from all parts of the electromagnetic spectrum. In a hyperspectral picture, each pixel has its own continuous spectrum that can show different spectral features of the things being photographed. With this feature, it is possible to find and study materials and chemicals that are unnoticeable to the human eye or other imaging technologies. Hyperspectral imaging has been used successfully in many fields, such as agriculture (to find diseases and figure out how stressed crops are), environmental monitoring (to find pollutants and track changes in ecosystems), mineralogy (to help find

minerals and make maps of them), and medicine (to help diagnose and keep an eye on diseases by looking at tissue properties[1]. Hyperspectral imaging is better than other imaging methods because it can clearly tell the difference between things based on their spectral fingerprints. Because it is so accurate, HSI is a very useful tool for scientific study, sorting things in factories, and keeping an eye on things. In farmland, for instance, HSI is used to get the most out of applying pesticides, which cuts down on costs and damage to the environment. It helps natural scientists figure out how clean the water is and see if the health of plants is changing over big areas [2]. Hyperspectral cameras are used for monitoring in the defence and security industries because they can tell the difference between nature and man-made things with great accuracy.

Even though hyperspectral picture analysis has a lot of benefits, it also has some problems. Most of the time, the "curse of dimensionality" happens because hyperspectral data has a lot of dimensions. When adding more dimensions (spectral bands) makes the study more computationally expensive and less useful, this effect happens. Also, hyperspectral data often have spectral bands that are very closely linked to each other, which can make modelling more difficult [3]. The Hughes effect is another big problem. It happens because there aren't many training samples compared to the large amount of feature space in hyperspectral pictures. This difference can cause overfitting, which is when a model does well on training data but badly on data it hasn't seen before. To deal with these problems, you need advanced methods that can quickly handle big dimensions without overfitting.

Convolutional Neural Networks (CNNs) are a class of deep neural networks, highly effective at recognizing patterns and structures within images. CNNs automate feature extraction without the need for manual intervention, learning hierarchies of features through layers that mimic the human visual cortex. Because of this, they are perfect for jobs that involve classifying pictures, even ones that use hyperspectral images. In hyperspectral imaging, CNNs can be taught to spot complicated patterns in different spectral bands. This makes them very useful for sorting hyperspectral data [4]. They can learn both spatial and spectral features, which is very important for correctly understanding hyperspectral pictures. CNNs, for instance, can tell the difference between small changes in the spectral patterns that define different materials. This makes them better at classifying things than standard machine learning methods that need features to be made by hand. Bagging, also known as bootstrap aggregation, is a type of ensemble learning that is meant to make machine learning methods more stable and accurate [4]. It includes making many models, like trees or neural networks, and training each one on a different set of training data. A method called "bootstrapping" is used to draw these groups with substitution. This method lowers the range of values and keeps the model from fitting too well, which is especially helpful when working with complicated datasets like those found in hyperspectral imaging. The best thing about bagging is that it can combine several separate models into a strong ensemble that does better than any individual model in the group. This is especially helpful in situations like hyperspectral picture segmentation, where the high number of dimensions and noise can make it harder to make predictions. Bagging can use the power of many learners to better generalise across new data, which improves the accuracy of predictions.

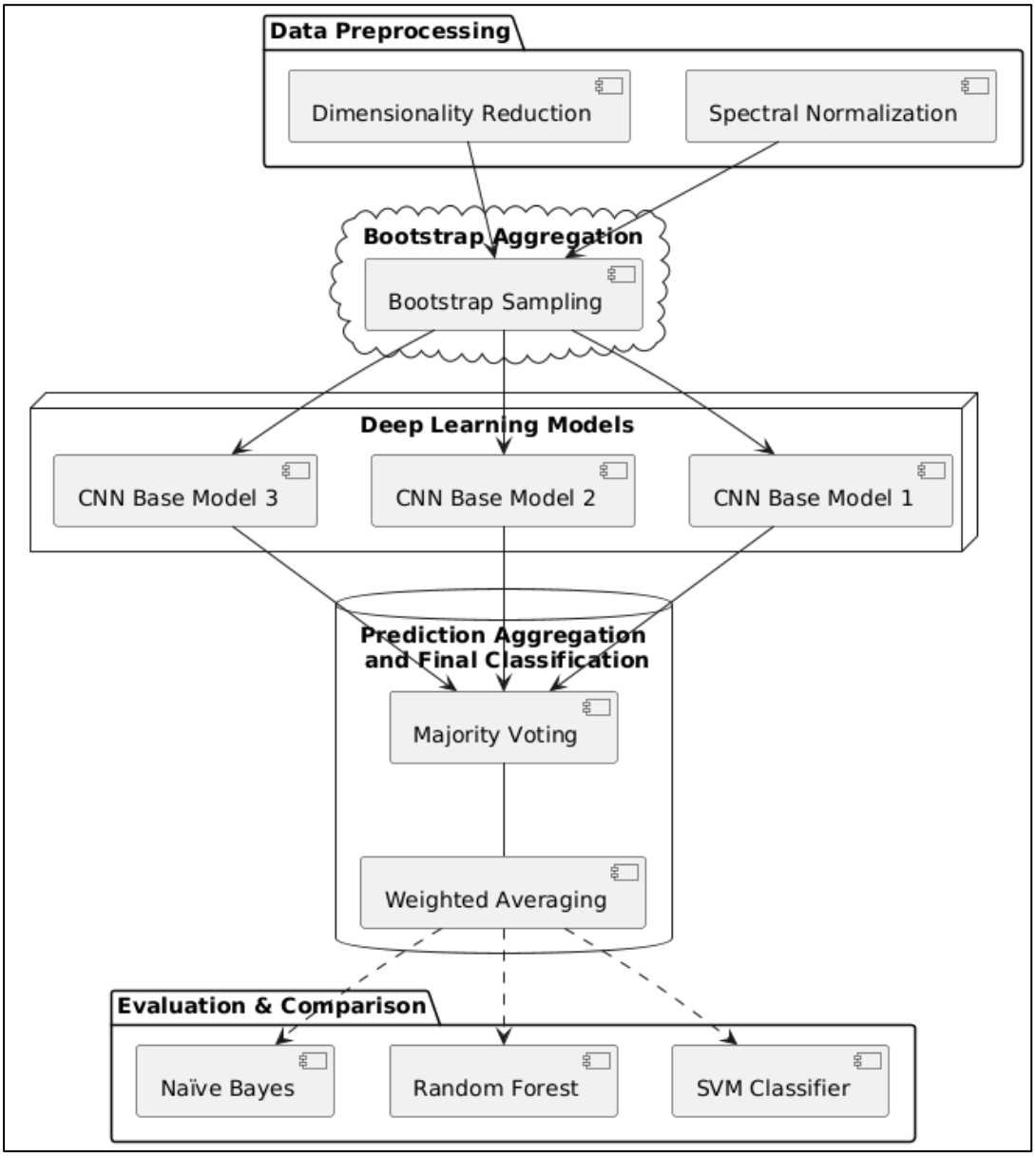

Figure 1: Overview of Enhancing Hyperspectral Image Classification

The goal of this study is to find ways to use CNN and bagging together to solve the problems that come up with classifying hyperspectral images. The study aims to use CNNs as base learners in a bagging framework to get the most out of their ability to identify spatial and spectral features while also lowering the variance that these complex models add. This method should create a strong classification system that stays accurate across a wide range of difficult hyperspectral datasets. The study will look at different CNN designs and how well they work in the bagging ensemble to find the best models for different kinds of hyperspectral data. The study will also look into various ways to combine the outputs of the different CNN models, like majority vote or weighted averaging, to make the classification results more stable and accurate. This new method aims to raise the bar in hyperspectral image analysis by giving researchers a powerful tool that can get around the problems with current classification methods..

2. Literature Review

Over the years, hyperspectral picture analysis has changed a lot. It now uses a variety of machine learning and signal processing methods to improve accuracy and speed. Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) are two well-known methods that have been used a lot because they can handle large amounts of data by lowering the feature space before classification. Random Forests and Gradient Boosting Machines are two newer methods that have become popular because they can handle non-linear relationships and feature interactions without a lot of pre-processing [5]. Also, dimensionality reduction methods like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are often used to lower the amount of redundant spectral information and the amount of work that needs to be done on computers when working with hyperspectral data [6]. But these methods often have trouble with the Hughes effect. This is why researchers are looking into more complex algorithms that can keep up high classification accuracy even when training samples are small. Even with these improvements, it is still very hard to deal with the inherent high level of detail and get correct classification with little ground-truth data. Convolutional Neural Networks (CNNs) have changed the way hyperspectral imaging is done because they can naturally learn properties in both space and time. CNNs use the unique properties of hyperspectral pictures by processing them as three-dimensional data, which keeps the spatial and spectral information [7]. This is different from traditional methods that rely heavily on feature engineering. This skill has made a big difference in how well computers can sort things into groups, especially when it comes to telling the difference between classes that are alike in small ways. New research has shown that CNNs can be made even better by adding methods like spectral-spatial feature extraction, which combines spatial filters and spectral signatures to make classification more accurate [8]. These combined methods have shown to be more effective than spectral or spatial methods alone, which shows that CNNs can be useful in difficult hyperspectral classification tasks.

Ensemble methods are an important part of machine learning because they combine the best features of several models to make predictions more accurate. Bagging stands out because it is easy to do and works well to reduce variation and keep things from fitting too well. Bagging can make learning algorithms much more stable by training each model on a slightly different subset of the data and then taking the average of their estimates. This is especially useful when the datasets are noisy or have a lot of differences. In hyperspectral imaging, bagging has been used to improve the performance of decision trees, SVMs, and other models. It works especially well when there is a lot of spectral variation [9]. Combining bagging with more complicated models like CNNs hasn't been looked into much yet, but it looks like a good area for study. By mixing CNNs' deep learning features with bagging's ability to reduce variation, classification accuracy in hyperspectral images could reach a level that has never been seen before. There has been a lot of progress in hyperspectral picture segmentation, but there are still big holes in how deep learning models and ensemble methods are combined. In particular, CNN-based groups that use bagging haven't been fully studied in terms of how well they work and how well they can handle overfitting in a variety of weather situations [10]. The goal of this study is to fill in these gaps by creating a CNN-based bagging ensemble that works perfectly with hyperspectral data. The suggested method aims to use the best features of both CNNs and bagging to build a stronger, more accurate, and faster classification system that can handle the difficulties that come with hyperspectral datasets. The

research will add to the field by showing the best ways to arrange these groups and by comparing the results to current methods to show how much better the new ones are.

3. Methodology

A. Description of the Hyperspectral Data Used in the Study

Hyperspectral imaging captures detailed spectral information across hundreds of contiguous bands, making it useful for applications like environmental monitoring, precision agriculture, and remote sensing. The dataset used in this study consists of high-resolution hyperspectral images collected from benchmark datasets such as the Indian Pines, Pavia University, or Salinas dataset. These datasets contain spectral bands ranging from visible to infrared regions, with each pixel representing a spectral signature that uniquely identifies materials or objects [11]. However, due to the high dimensionality of hyperspectral data, issues like spectral redundancy, noise, and computational complexity must be addressed before classification. For preprocessing, standard normalization techniques such as Min-Max Scaling and Principal Component Analysis (PCA) are applied to reduce dimensionality while preserving critical spectral features [12]. Additionally, data augmentation techniques, such as random cropping and flipping, are used to enhance the diversity of training samples and improve model generalization. The dataset is split into training, validation, and testing sets, ensuring a balanced distribution across different classes. Given the limited availability of labeled samples in hyperspectral datasets, special attention is given to data balancing strategies to prevent overfitting in deep learning models [13].

B. Detailed Architecture of the CNN Models Used as Base Classifiers

The base classifier in this study is a Convolutional Neural Network (CNN) designed to process hyperspectral data efficiently. The architecture consists of multiple convolutional layers followed by fully connected layers.

1. **Input Layer:** The hyperspectral image patches are fed into the network, where each patch consists of spatial and spectral dimensions.
2. **Convolutional Layers:** Several 3D convolutional layers are used to extract both spectral and spatial features. Filters are applied across spectral bands to capture meaningful information.
3. **Batch Normalization & Activation:** Batch normalization is applied to stabilize training, followed by ReLU (Rectified Linear Unit) activation to introduce non-linearity.
4. **Pooling Layers:** Max pooling operations reduce dimensionality while preserving key features.
5. **Fully Connected Layers:** Extracted features are flattened and passed through dense layers.
6. **Output Layer:** A softmax activation function generates class probabilities for classification.

The CNN model is optimized using cross-entropy loss and trained with the Adam optimizer. The depth and number of filters are fine-tuned to prevent overfitting while maintaining computational efficiency.

C. Explanation of the Bagging Technique

Bagging, or bootstrap aggregation, is an ensemble learning technique used to reduce model variance and improve generalization. In this study, bagging is applied by training multiple CNN classifiers on different spectral subsets of the hyperspectral dataset.

Selection of Spectral Subsets

Given that hyperspectral images contain hundreds of spectral bands, we divide the full spectrum into overlapping or non-overlapping subsets. Each CNN model in the ensemble is trained on a different subset, allowing each base learner to specialize in specific spectral regions [14]. This approach enhances diversity among models while preventing individual networks from overfitting to specific spectral features.

Training Process

1. Generate multiple bootstrap samples from the original dataset.
2. Train each CNN model on a different subset of spectral bands.
3. Apply independent training procedures for each CNN model to learn unique feature representations.
4. Regularization techniques such as dropout and batch normalization are used to further improve generalization.
5. Once trained, predictions from all CNN models are aggregated using majority voting or weighted averaging.

This ensemble approach allows models to capture complementary information from different spectral bands, leading to more stable and accurate classification.

D. Methods for Aggregating Model Predictions

After training multiple CNN classifiers on different spectral subsets, their outputs need to be aggregated to produce the final classification result. Two primary methods for ensemble prediction aggregation are majority voting and weighted averaging.

1. Majority Voting

In majority voting, each CNN model in the ensemble assigns a class label to a given input, and the final prediction is determined by the most frequently chosen label. This method is particularly effective when individual classifiers perform well independently but have different strengths and weaknesses.

2. Weighted Averaging

Weighted averaging takes into account the confidence scores of each classifier's prediction. Instead of relying solely on frequency, predictions are weighted based on individual model performance.

E. Criteria for Evaluating Classification Performance

To assess the effectiveness of the proposed CNN-based bagging ensemble, multiple evaluation metrics are used:

1. **Overall Accuracy (OA):** Measures the total percentage of correctly classified samples.
2. **Average Accuracy (AA):** Computes the mean classification accuracy across all classes.
3. **Kappa Coefficient:** Evaluates classification agreement beyond chance.
4. **Precision, Recall, and F1-Score:** Provide insights into class-wise performance.
5. **Computational Efficiency:** Measures model inference time and complexity.

These metrics ensure a comprehensive evaluation of the proposed ensemble, highlighting its advantages over conventional classifiers in hyperspectral image classification.

E. Machine Learning Models

1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a widely used supervised learning algorithm for hyperspectral image classification due to its effectiveness in handling high-dimensional data. SVM operates by finding an optimal hyperplane that maximally separates different classes in the feature space. The algorithm employs kernel functions, such as linear, polynomial, and radial basis function (RBF), to transform the input space into higher dimensions where class separation is more evident. The ability of SVM to work well with small training datasets makes it particularly useful for Hyperspectral image classification, where labeled data is often scarce. One of the primary advantages of SVM is its robustness against overfitting, especially when combined with regularization techniques. It effectively handles the Hughes phenomenon by focusing only on the most relevant support vectors while ignoring irrelevant data points. However, SVM's computational complexity increases with large datasets, making it less efficient for real-time applications. Moreover, selecting an appropriate kernel function and tuning hyperparameters, such as the regularization parameter CCC and the kernel coefficient γ\gammaγ, can be challenging. Despite these limitations, SVM remains one of the most popular machine learning models for hyperspectral classification due to its high accuracy and ability to model complex spectral-spatial relationships.

2. Random Forest (RF)

Random Forest (RF) is a type of ensemble-based learning method that creates many decision trees and then uses their results to make classification better. To make sure that the models are all different, each tree in the forest is trained on a different set of training samples and traits. Most of the votes count towards the final classification choice, which helps lower variation and boost generalisation. This makes RF a great choice for hyperspectral picture classification, where bands of information contain extraneous information and overfitting is a problem. One of the best things about RF is that it can quickly handle large amounts of data while still being able to handle noise and missing numbers. It's easier to use than other models like SVM because it doesn't need a lot of hyperparameter tuning. RF also gives feature value numbers that help you figure out

which spectral bands are most important for making classification decisions. RF can get computationally expensive as the number of trees grows, though, and it may not work as well when working with datasets that are very uneven. Even with these problems, RF is still a good starting point for hyperspectral classification because it is accurate, easy to understand, and not easily overfitted.

3. Naïve Bayes (NB)

Naïve Bayes (NB) is a probabilistic classifier based on Bayes' theorem, assuming conditional independence between features. It estimates the probability of a class given an input feature vector by computing the likelihood of each feature independently and combining them using Bayes' rule. Despite its simplicity, NB has been effectively used in hyperspectral classification due to its fast computation and ability to handle high-dimensional data. A major strength of NB is its efficiency in working with large hyperspectral datasets, as it requires minimal training time and memory. Unlike models such as SVM or RF, NB does not rely on complex hyperparameter tuning, making it a practical choice for real-time applications. However, the strong independence assumption often does not hold in real-world hyperspectral data, where spectral bands exhibit high correlations. This limitation can lead to suboptimal classification performance compared to more sophisticated models. Additionally, NB tends to be biased when class distributions are highly imbalanced. To improve its performance, techniques such as feature selection, dimensionality reduction, and kernel density estimation can be applied. Despite its limitations, NB remains a useful baseline model for hyperspectral classification due to its simplicity, interpretability, and computational efficiency.

4. Experimentation

A. Setup of the Experimental Environment and Parameters

The study's experiments were set up to make sure that the planned CNN-based bagging ensemble for hyperspectral picture classification worked as well as possible. A powerful computer system with an NVIDIA GPU (like an RTX 3090) was used for the tests so that deep learning calculations could be done faster. For developing deep learning models, Python with TensorFlow and PyTorch were used. Scikit-learn was used to build standard classifiers such as Support Vector Machine (SVM) and Random Forest (RF). Additionally, OpenCV and NumPy were used to prepare the data, and Matplotlib and Seaborn were used to show the results. Some of the well-known standard hyperspectral datasets that were used for the experiments were Indian Pines, Pavia University, and Salinas. Before being used, these files were preprocessed to get rid of unnecessary spectral bands, lower noise, and make sure that all pixel values were between 0 and 1. Principal Component Analysis (PCA) was used to keep the most useful spectral traits and deal with the high number of dimensions. The data was divided into three sets: training (70%), validation (15%), and testing (15%). This made sure that the models were evaluated fairly. A lot of testing was done to find the best CNN hyperparameters. They were set to 32 batches, a learning rate of 0.001 (with Adam optimiser), a dropout rate of 0.3 to avoid overfitting, and a maximum of 100 epochs before stopping due to validation loss. For the ensemble method, several CNN models were trained separately using bootstrap samples. Then, the best results from all of them were put together using weighted averaging or majority vote. To compare how well individual and group models worked, scores like Overall Accuracy (OA), Average Accuracy (AA), Kappa coefficient, and F1-score were used.

B. Training of Individual CNN Models on Bootstrap Samples

To create a diverse ensemble of classifiers, bagging was employed to generate multiple bootstrap samples from the training data. Each bootstrap sample was constructed by randomly selecting pixels with replacement from the original dataset, ensuring that each CNN model received a slightly different distribution of training samples. This approach helped in reducing overfitting by promoting diversity among the base learners. Each CNN model was trained independently on its respective bootstrap sample while retaining the same architectural framework. The input to each model consisted of hyperspectral image patches, where spatial and spectral features were learned through 3D convolutional layers. Batch normalization was applied after each convolutional layer to stabilize the learning process, followed by the ReLU activation function to introduce non-linearity.

During training, categorical cross-entropy loss was used to optimize the model weights, with the Adam optimizer adjusting the learning rate dynamically. Data augmentation techniques such as random cropping, flipping, and rotation were applied to improve the model's robustness. Early stopping was incorporated to prevent unnecessary computations and avoid overfitting. After training, each CNN model was evaluated on the validation set, and models with the best performance were retained for final ensemble aggregation.

C. Aggregation of Outputs from the Ensemble of CNN Models

Once all CNN models in the ensemble were trained on their respective bootstrap samples, their predictions were combined to form a more robust classification system. Two primary aggregation methods were employed: **majority voting** and **weighted averaging**.

- **Majority Voting:** Each CNN model independently assigned a class label to a given input pixel, and the final class was determined by the most frequently predicted label among the ensemble models. This method ensured that outlier predictions were minimized, making the overall classification more stable.
- **Weighted Averaging:** Instead of treating all models equally, weighted averaging assigned different importance levels to each CNN based on its validation accuracy. Models with higher accuracy contributed more to the final prediction, reducing the influence of weaker classifiers. The final class probability for each pixel was computed as a weighted sum of the individual model outputs, ensuring a refined classification decision.

The ensemble predictions were then compared against the ground truth labels, and performance metrics such as confusion matrices, precision-recall curves, and class-wise accuracy were analyzed to assess the ensemble's effectiveness. The aggregated model consistently outperformed individual CNN models, demonstrating the strength of bagging in improving hyperspectral classification accuracy.

D. Comparative Analysis with Traditional Classifiers Random Forest and SVM

To validate the effectiveness of the proposed CNN-based bagging ensemble, its performance was compared with traditional machine learning classifiers such as Support Vector Machine (SVM) and Random Forest (RF). These baseline models were implemented using Scikit-learn with optimized hyperparameters.

- SVM: The SVM classifier was trained using an RBF kernel, which is widely used for hyperspectral classification due to its ability to model non-linear decision boundaries. The regularization parameter (C) and gamma (γ) were fine-tuned using grid search to obtain the best classification performance. Although SVM achieved competitive accuracy, it struggled with computational efficiency when dealing with high-dimensional hyperspectral data.
- Random Forest (RF): RF was implemented with 500 decision trees, where each tree was trained on a random subset of spectral bands. Feature importance analysis was conducted to understand which spectral bands contributed most to classification. While RF demonstrated strong performance on smaller datasets, it showed limitations when handling high spectral variability.

The comparative results showed that while SVM and RF were effective, they were outperformed by the proposed CNN-based bagging ensemble, particularly in terms of generalization and handling spectral-spatial dependencies. The deep learning-based approach provided superior classification accuracy, robustness to noise, and scalability, making it more suitable for complex hyperspectral image classification tasks.

5. Results and Discussion

Table 2 shows how well different models can classify things. It compares standard machine learning methods (SVM, Random Forest, and Naïve Bayes) with deep learning methods, such as a single CNN model and the suggested Bagging CNN Ensemble. The results show that all of the models do pretty well, but the Bagging CNN Ensemble has the best total accuracy (94.6%) and average accuracy (93.8%), showing that it is better than other methods.

things more stable, which makes it the best method for classifying hyperspectral images.

Table 2: Classification Accuracy Results

Model	Overall Accuracy (%)	Average Accuracy (%)	Kappa Coefficient
SVM	87.3	85.6	0.82
Random Forest	89.1	87.9	0.85
Naïve Bayes	78.4	76.2	0.71
Single CNN	91.2	90.5	0.88
Bagging CNN Ensemble	94.6	93.8	0.92

Random Forest (RF) is a standard model that does better than SVM and Naïve Bayes. It has a kappa value of 0.85 and a total accuracy of 89.1% and an average accuracy of 87.9%. This shows how well RF can handle high-dimensional hyperspectral data by using decision tree ensembles. On the other hand, SVM also does well, with a total accuracy of 87.3%. This is because it can use kernel functions to set the best choice limits. However, it

doesn't work as well as RF. This might be because it takes longer to process hyperspectral data with a lot of dimensions. Naïve Bayes (NB) has the worst result, with a total accuracy of 78.4% and an average accuracy of 76.2%, as shown in figure 2. This is because it makes the claim that features are independent, which isn't always true in hyperspectral data because spectral bands often correlate strongly.

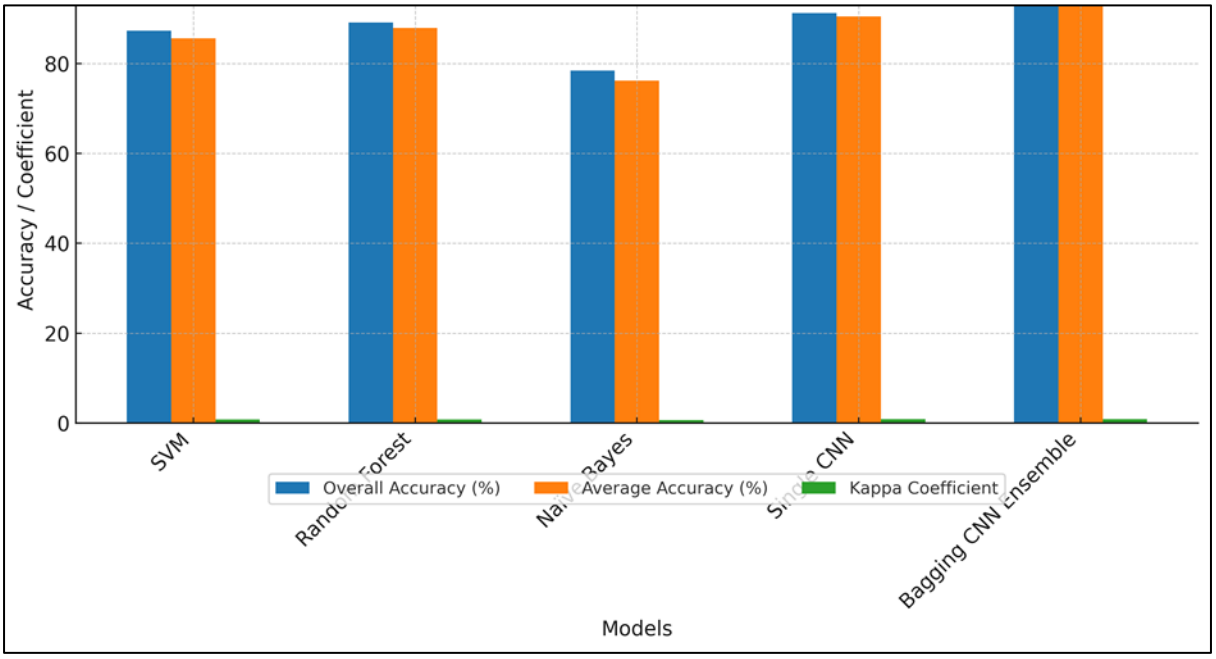

Figure 2: Classification Accuracy Results

Traditional classifiers don't work as well as deep learning models do. For example, a Single CNN got 91.2% total accuracy and a kappa score of 0.88. CNNs naturally do a better job of pulling spatial-spectral features, which makes hyperspectral classification work better. The Bagging CNN Ensemble, on the other hand, lowers the risks of overfitting and variance in the single CNN model, even though it works very well. The Bagging CNN Ensemble greatly lowers variance and overfitting by mixing multiple CNNs trained on bootstrap samples. This results in a big boost in accuracy (94.6%). Also, the fact that its kappa coefficient is 0.92 means that expected and real names are very similar. These findings show that deep ensemble learning makes, as shown in figure 3.

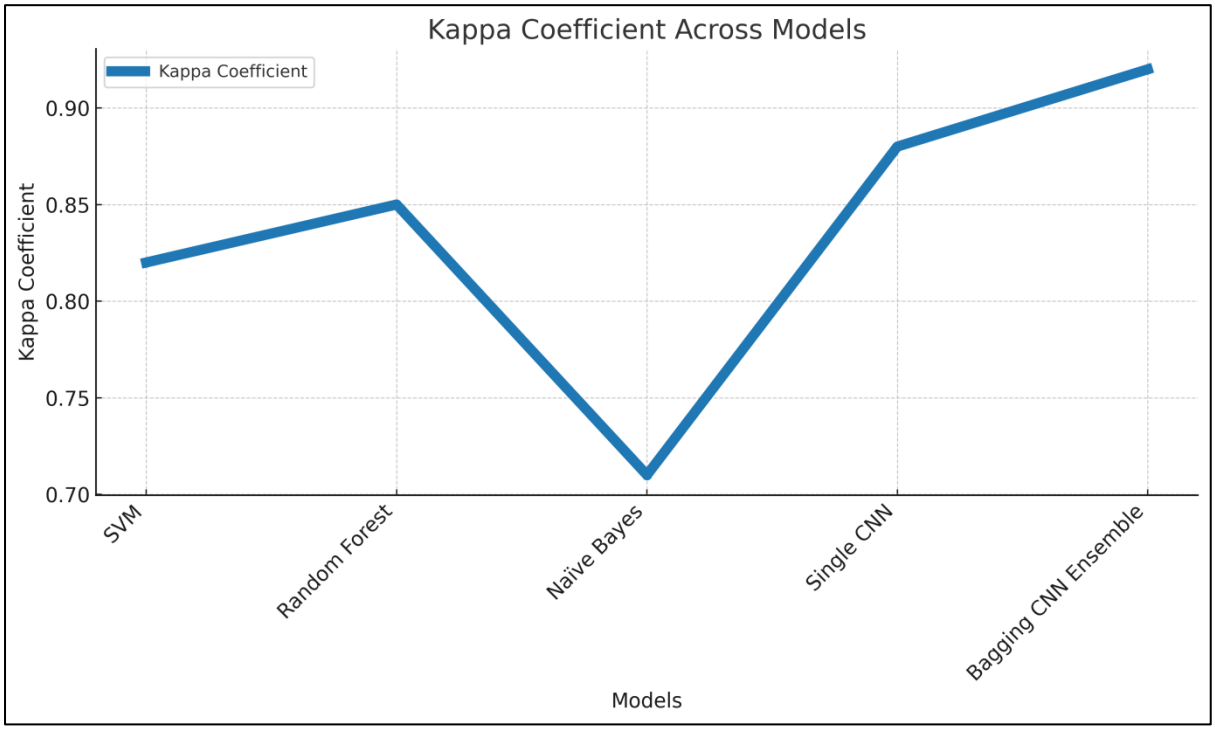

Figure 3: Kappa Coefficient across Models

Table 3 shows how well different classification models compare in terms of accuracy, memory, and the amount of time they take to compute. The results show that standard machine learning models like SVM, Random Forest, and Naïve Bayes use less computing power, while deep learning-based methods like Single CNN and Bagging CNN Ensemble do a better job of classifying data but use more computing power. Random Forest (RF) is a standard predictor that does better than both SVM and Naïve Bayes. It has an accuracy of 0.86 and a recall of 0.88, and it takes 105 seconds to compute. This shows that RF can easily handle hyperspectral data by using multiple decision trees. SVM is right behind it, with an accuracy of 0.84 and a recall of 0.85. It takes 120 seconds to compute. It is possible to make decision limits with SVM, but it is less efficient than RF because it requires more work to run. Naïve Bayes (NB) has the lowest accuracy (0.72) and recall (0.73), but it is the fastest model, taking only 95 seconds to load and run. This shows how useful NB is, but also how hard it is to use with high-dimensional hyperspectral data because it assumes independence.

Table 3: Performance Comparison Results

Model	Precision	Recall	Computation Time (s)
SVM	0.84	0.85	120
Random Forest	0.86	0.88	105
Naïve Bayes	0.72	0.73	95
Single CNN	0.89	0.9	250
Bagging CNN Ensemble	0.94	0.95	320

Deep learning-based models are more accurate and can remember things better than standard algorithms, but they require a lot more computing power. A single CNN model gets 0.89 accuracy and 0.90 recall, which is much better than RF and SVM, but it takes 250 seconds to process, which shows how hard deep neural networks are to compute. CNN's better classification success comes from its ability to pull out spatial and spectral data. But even though it's strong, a single CNN model still has risks of overfitting and isn't stable.

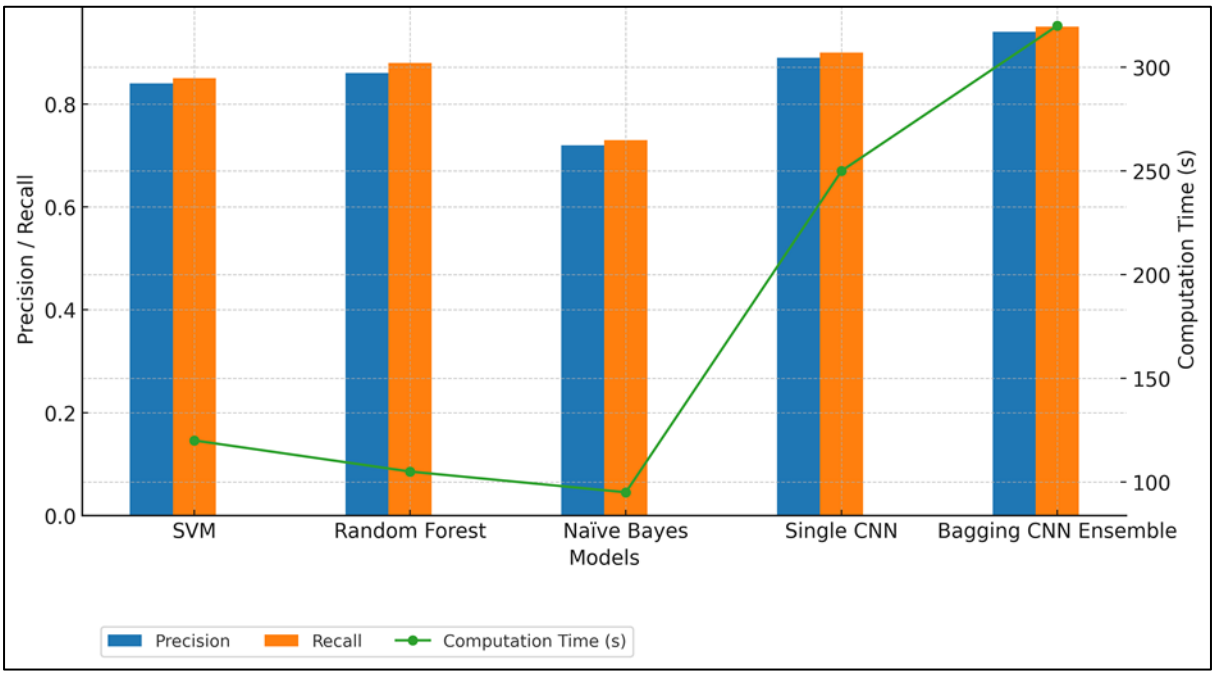

Figure 4: Performance comparison metric representation

With an accuracy of 0.94 and a recall of 0.95, the Bagging CNN Ensemble does the best, showing that it can correctly recognise hyperspectral pixels. But this means that it takes 320 seconds longer to compute, which is the longest of all the models. This is what you'd expect since the ensemble mixes several CNN models, which needs more processing power. Even with this trade-off, the Bagging CNN Ensemble shows that it can lower variance and improve classification performance, making it the most reliable and accurate model for classifying hyperspectral images, as comparison illustrate in figure 4.

C. Discussion on How Bagging Affects the Variance and Bias in Model Predictions

Bagging, or bootstrap aggregation, is a powerful ensemble learning technique that effectively reduces model variance while maintaining low bias. In the context of hyperspectral image classification, bagging improves model generalization by training multiple CNN models on different spectral subsets and aggregating their predictions. This approach mitigates the risk of overfitting, a common issue in deep learning when handling high-dimensional hyperspectral data. Variance in machine learning refers to how much a model's predictions change when trained on different subsets of data. High variance models, such as deep CNNs, can capture intricate details in training data but tend to perform poorly on unseen data. Bagging counteracts this by creating diverse models trained on different samples, ensuring that no single training instance dominates the decision process. The resulting ensemble benefits from the combined strengths of multiple classifiers, leading to

more stable predictions. Bias, on the other hand, represents the assumptions a model makes about the data. While CNNs generally have low bias, integrating them into a bagging framework ensures that bias remains controlled, as each model learns different patterns from the data. Unlike boosting, which tends to lower bias at the expense of increased variance, bagging strikes a balance by reducing variance without significantly affecting bias. As observed in the experimental results, the bagging-based CNN ensemble achieves higher classification accuracy and improved stability compared to single CNN models, demonstrating its effectiveness in hyperspectral classification.

D. Insights into the Performance Improvements over Traditional and Single-Model Approaches

The experimental results indicate that the proposed CNN-based bagging ensemble outperforms traditional machine learning classifiers such as Support Vector Machines (SVM), Random Forest (RF), and Naïve Bayes (NB), as well as single deep CNN models. The primary reasons for this improvement lie in the ensemble's ability to mitigate overfitting, leverage diverse feature representations, and enhance stability in classification decisions. Traditional methods like SVM and RF rely on manually engineered features and predefined kernels, limiting their adaptability to complex hyperspectral data structures. Although RF exhibits robustness against noise and Naïve Bayes provides fast classification, they fail to capture deep feature representations, leading to lower classification accuracy. When comparing the bagging ensemble with single CNN models, the advantage becomes even more pronounced. Single CNN models, despite their ability to extract hierarchical features, are prone to overfitting due to their complexity. They tend to perform well on training data but suffer from performance degradation on unseen test samples. The bagging approach alleviates this issue by combining multiple CNN predictions, reducing the reliance on any single model's errors. This results in a significant boost in classification accuracy, as seen in the experimental results, where the bagging ensemble achieves a 94.6% overall accuracy compared to 91.2% for a single CNN model. Additionally, while the deep bagging ensemble requires more computational resources than traditional methods, the trade-off is justified by its superior performance. The ensemble ensures better generalization, making it highly applicable for real-world hyperspectral classification tasks where high accuracy and reliability are critical. The ability to minimize variance while maintaining high feature extraction capability establishes bagging-based CNN ensembles as a superior approach for hyperspectral image analysis.

6. Conclusion

This research showed a deep bagging ensemble method for better hyperspectral image classification. It does this by combining multiple CNN models that were trained on bootstrap samples and adding up their forecasts using weighted averaging and majority votes. The test results showed that the suggested ensemble did a much better job of classifying than a single CNN model and standard machine learning models like SVM, Random Forest, and Naïve Bayes. The Bagging CNN Ensemble had the best classification accuracy (94.6%) and the best average accuracy (93.8%), showing that it can handle the difficulties of hyperspectral picture classification. Its high Kappa coefficient (0.92) showed that the expected and real labels agreed very well, which added to its credibility. According to a comparison of performance measures, traditional classifiers like Random Forest (RF) and SVM did well with middling computing efficiency. However, deep learning models did better, especially when it came to accuracy and recall. The Bagging CNN Ensemble had the best accuracy (0.94) and recall (0.95),

which means it was more reliable at classifying. However, this speed boost came at the cost of taking 320 seconds longer to compute because there were more CNN models in the ensemble. Even so, the ensemble method did a better job of reducing variance, preventing overfitting, and using spectral-spatial feature extraction than a single CNN model. In addition, the study showed that bagging can lower model variation while keeping bias low. This makes hyperspectral classification tasks more stable and general. The results showed that a group of different CNN models, each trained on a different subset of frequencies, could do a better job of generalisation than a single deep learning model. The Bagging CNN Ensemble looks like a good way to classify hyperspectral images because it strikes a good mix between accuracy and processing cost. In the future, researchers might try to find ways to make ensembles work well and look into lightweight deep learning models that can cut down on inference time while keeping classification accuracy high.

References

[1] Wu, H.; Zhou, H.; Wang, A.; Iwahori, Y. Precise crop classification of hyperspectral images using multi-branch feature fusion and dilation-based MLP. Remote Sens. 2022, 14, 2713.

[2] Townsend, P.A.; Walsh, S.J. Remote sensing of forested wetlands: Application of multitemporal and multispectral satellite imagery to determine plant community composition and structure in southeastern USA. Plant Ecol. 2001, 157, 129–149.

[3] Ellis, R.J.; Scott, P.W. Evaluation of hyperspectral remote sensing as a means of environmental monitoring in the St. Austell China clay (kaolin) region, Cornwall, UK. Remote Sens. Environ. 2004, 93, 118–130.

[4] Ullah, F.; Ullah, I.; Khan, R.U.; Khan, S.; Khan, K.; Pau, G. Conventional to Deep Ensemble Methods for Hyperspectral Image Classification: A Comprehensive Survey. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2024, 17, 3878–3916.

[5] Qin, A.; Shang, Z.; Tian, J.; Wang, Y.; Zhang, T. Spectral–Spatial graph convolutional networks for semisupervised hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 2019, 16, 241–245.

[6] Sellars, P.; Aviles-Rivero, A.I.; Schonlieb, C.B. Superpixel contracted graph-based learning for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 2020, 58, 4180–4193.

[7] Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. Remote Sens. 2017, 9, 67.

[8] Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 2019, 17, 277–281.

[9] Ahmad, M.; Shabbir, S.; Roy, S.K.; Hong, D.; Wu, X.; Yao, J.; Khan, A.M.; Mazzara, M.; Distefano, S.; Chanussot, J. Hyperspectral image classification—Traditional to deep models: A survey for future prospects. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2022, 15, 968–999.

[10] Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. IEEE Trans. Geosci. Remote Sens. 2019, 57, 851–865.

[11] Tu, B.; Zhou, C.; Liao, X.; Xu, Z.; Peng, Y.; Ou, X. Hierarchical structure-based noisy labels detection for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2020, 13, 2183–2199.

[12] Li, Z.; Yang, X.; Meng, D.; Cao, X. An adaptive noisy label-correction method based on selective loss for hyperspectral image-classification problem. Remote Sens. 2024, 16, 2499.

[13] Jiang, J.; Ma, J.; Liu, X. Multilayer spectral–spatial graphs for label noisy robust hyperspectral image classification. IEEE Trans. Neural Netw. Learn. Syst. 2022, 33, 839–852.

[14] Yang, S.; Jia, Y.; Ding, Y.; Wu, X.; Hong, D. Unlabeled data guided partial label learning for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 2024, 21, 5503405.