Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Forest Fire Image Classification via Hybrid Deep Learning and Stacking Ensemble Technique

Israa Ali Alshabeeb*

Department of Computer Networks and Software Techniques, Babylon Technical Institute, Al-Furat Al-Awsat Technical University, Babil, Iraq. Email: Inb.esr@atu.edu.iq

ARTICLE INFO

Article history:
Received: 04/08/2025
Rrevised form: 28/08/2025
Accepted : 31/08/2025
Available online: 30/09/2025

Keywords:

Forest Fire Classification,
Image Embedding,
Hybrid Deep Learning,
Stacking Ensemble,
Pre-trained Models,
Machine learning

ABSTRACT

In recent years, forest fires have increased significantly and in an intimidating manner worldwide, thereby affecting both the environment and human life. When they are overlooked and not detected at an early stage and in a short period of time, they will spread rapidly. Accurate and early fire detection and classification are crucial for disaster control and well-timed emergencies. To address this problem, a hybrid model of pre-trained embedding and diverse classifiers to extract features, detect, and classify fire images with optimized performance and stacking with different meta-learners to improve the reliability, is applied in this paper. Three types of pre-trained models, including InceptionV3, SqueezeNet, and DeepLoc, were used for feature extraction with different classifiers: Neural Network (NN), Decision Tree (DT), and XGBoost. The proposed model, which involves stacking multiple classifiers, provides an accurate, optimal, and efficient response for fire classification. A total of 1520 images were used in this study. The best performance was achieved by integrating InceptionV3 with NN, XGBoost, and Logistic Regression as a meta-learner, yielding 98.9% accuracy, 99.9% AUC, and 98.9% for each of (F1-score, Precision, and Recall) with 97.8% MCC, with 10 False Negatives and 7 False Positives in error types. This combination of deep and machine learning with stacking shows consistent progress across all experiments, and these results can enhance and improve the quality, leading to more dependable detection and classification in safety-critical applications.

https://doi.org/10.29304/jqcsm.2025.17.32402

1. Introduction

Wildfires, especially forest fires, are damaging phenomena that can critically affect and alter the Earth's ecosystem balance [1]. According to a recent study, approximately 1.009 billion people are affected and exposed to wildfire smoke and air pollution in indoor environments at least one day annually [2]. Multiple volatile organic compounds caused by smoke remained for several days, leading to exposure of individuals to chemical air pollutants indoors [3]. Moreover, fires change soil characteristics and lead to increased erosion processes and decreased soil infiltration. These changes have significant effects over an extended period on water quality, base flow, and groundwater. Pollutants caused by fires can harm public health and aquatic ecosystems [4].

The harmful effects of forest fire, such as climate change and the acceleration of global warming, highlight the imperative need for advanced detection systems [5]. Traditional ways for detecting forest fires were manual. However, these systems and methods are often ineffective and prone to errors due to human factors. Furthermore, convolution sensors used to detect different cases, such as smoke, flame, or heat, usually suffer from delays and limitations in coverage. Therefore, the need to develop and implement many units has become increasingly

∗Corresponding author: Israa Ali Alshabeeb

Email addresses: inb.esr@atu.edu.iq

Communicated by ‘sub etitor’

important for effective and efficient detection and monitoring [6]. The rapid and initial reliable detection of fire-wilds is essential to reduce the disasters [1]. Accurate classification and detection of fire images remain a critical concern due to the strong resemblance of non-fire images and changes in lighting. Recent research using various machine learning and deep learning models has focused on improving the detection and classification of forest fire images.

Many approaches have been suggested to classify and detect forest fire and non-fire images. Using VGG19 with the DeepFire dataset and reaching 95% of accuracy for the classification of fire and non-fire images [7]. While [8]proposed a ForestResNet model based on ResNet50 for detecting forest images with an accuracy of 92%. In addition, RCNN was utilized for forest fire classification from video frames and achieved 97.29% accuracy [9]. The DeepFire dataset has been used for the forest fire classification task. One study employed the FFireNet model for fire classification in smart cities and obtained 98.42% accuracy [10]. While another study applied EfficientNetB7, ACNet, and Bayesian optimization for fire classification and achieved 95.97% accuracy when using the DeepFire dataset and 97.45% accuracy for the FLAME dataset [11]. A combination of traditional image processing methods with CNN and an adaptive pooling technique was proposed in [12] to improve forest fire recognition. However, PSO based on Federated Learning was introduced in [13] for forest fire detection. A developed deep learning model based on Yolov5 was implemented to enhance the performance of detecting forest fires [14]. Another study employed RBFN-RAISR and InceptionV3 for accurate and fast fire detection, achieving 97.55% accuracy [15]. Both IoT sensors and the J48 algorithm for detecting forest fire images in Thailand. The model achieves 72% of accuracy for training data and 69% for the validation data [16]. In another study, a developer introduced a fire image detection system using Detectron2, Autodistill-based knowledge distillation, and Detection Transformer. YOLOv8n achieved the best performance with 95.21% accuracy and 98.5% of F1 score [17]. Enhancing the wildfire detection system using satellite imagery was introduced. Comparing three models: U-net, CNN, and autoencoder, which demonstrates that the CNN achieved more effective performance with 82% accuracy [18]. Three CNN models (VGG-16, InceptionV3, and ResNet50) were combined to detect and classify forest fires in aerial images. This combined approach attained 95.8% of accuracy [19]. The model incorporates RNN, PBNN, and LSTM using meteorological values as input to establish predictive models with the highest accuracy, with LSTM achieving an accuracy of 90.9%. The results indicate that using meteorological data for predicting wildfires with high accuracy [20]. In a recent study [21], the authors developed a stacking model for predicting forest fire risk using topographic, meteorological, human activity features, and vegetation data. They applied four ML models, which are random forest, LightGBM, MLP, and XGBoost, to combine in stacking task. Their results show that the stacking of traditional methods can increase and improve the prediction accuracy of wildfire risk with 97% AUC. Another study combined CNN with ResNet in stacking to classify image disasters such as floods, volcanoes, earthquakes, and wildfires. XGBoost was applied as a single meta-learner. The results of this combination obtained an improvement of the overall performance of classification with 95% accuracy [22]. A more recent work compared multiple pre-trained techniques to classify weather images. The pre-trained techniques included SqueezeNet, VGG16, VGG19, Inception V3, Pinters, and DeepLoc, while the classifiers were NN, SVM, and KNN. InceptionV3 with NN achieved the highest performance with 96.1%, highlighting the importance of selecting the best models for weather classification [23].

While many techniques have been used for classifying disaster images and providing valuable insights, the combination of using multiple deep learning models with diverse machine learning methods with stacking ensemble has not been explored enough and examined in depth, so there is still room for improvement, especially when using different meta-learners for the classification of the forest fire images. Therefore, a clear gap exists in this integration for testing this task, which could improve performance and make outcomes more consistent and reliable.

The objectives of this research paper are:

1. Explore and compare multiple pre-trained techniques for feature extraction (InceptionV3, SqueezeNet, and DeepLoc).
2. Apply and evaluate the performance of different ML classifiers.
3. Build a stacking ensemble with diverse models and various meta-learners to explore configurations that improve the performance of classification.
4. Identify the most effective model and evaluate the performance of the proposed and developed approach using the standard metrics.

This study contributes to the field of forest fire image classification and detection by applying a hybrid model that integrates deep learning for feature extraction and machine learning for training and classification, utilizing multiple diverse models with different meta-learners. Demonstrated that the stacking approach improves the performance for classification compared to the individual classifiers. Furthermore, it shows how the selection of pre-trained and meta-learners affects the results. This will highlight the effectiveness of integrating heterogeneous learners. These findings help to provide practical value for optimizing and developing AI models for early wildfire detection and environmental emergency mitigation.

This work provides values at both theoretical and practical levels. Theoretically, illustrates that the integration of pre-trained models with a stacking ensemble improves and enhances the performance of fire classification. Practically, the proposed model provided a dependable method for environmental monitoring by supporting reliable forest fire classification. This can help with early warning systems and optimize the risk management of wildfires.

The rest of this paper is organized as follows: Section 2 provides a background overview of the techniques used in this work. Section 3 describes the materials and methods, while Section 4 presents the results and discussion. Finally, section 5 concludes the work with recommendations for future work.

2. Technical Background

This section offers an overview of the pre-trained CNN models and classification algorithms used in this study.

2.1. Deep Learning Techniques

2.1.1. InceptionV3

This model is an improved version of InceptionV1 using multiple techniques to enhance the network with a deeper architecture than InceptionV1 and V2. It is considered a deep convolutional neural network and can be trained on low-specification systems, which can take many days to train. To solve this problem, transfer learning is used for this purpose to improve it [24].This model is another name of the GoogLeNet model and is composed of 42 layers with different sizes: 1×1, 3×3, and 5×5 convolutions. This architecture makes it more effective, including reducing the size of the grid, decomposition into reduced convolutions, and using auxiliary classifiers [25].

2.1.2. SqueezeNet

SqueezeNet introduced by Landola et all. It is a lightweight CNN that reaches AlexNet performance by reducing the number of parameters by 50 times on the ImageNet dataset [26]. It employs 1×1 conversions and fire modules to expand layers and reduce parameter count, making it ideal for contexts with vital memory limitations [27]. There are three key strategies for this model to reduce the complexity: minimizing 3×3 convolutional filters to 1×1 filters, reducing the number of input channels to 3×3, and the network delays down-sampling to preserve greater resolution activation maps [26].

2.1.3. DeepLoc

The third embedder used from the Orange tool in this paper was DeepLoc [28] [29].

2.2. Machine Learning Models

2.2.1. Decision Tree

Another popular model for classification and regression in data mining supervised ML is the decision tree, which operates by partitioning the data into many subsets according to attribute values, resulting in a decision model organized in a hierarchical structure [30]. It effectively processes both numerical and categorical data with minimal time consumption, delivering strong performance for large datasets with the ability to visualize the classification outcomes and to demonstrate relationships among predictors [31].

2.2.2. Neural Network

An artificial neural network (ANNs), like the Multilayer Perception (MLP), is a flexible and intelligent model that is inspired by the biological nervous system. It has a high ability to model both linear and nonlinear features and capture complex patterns within the dataset. Over the past three decades, NNs have been widely used in solving tasks such as prediction problems, classification, regression, and pattern regression events. In addition, the effectiveness of an ANN depends on how well it learns from data. A feed-forward neural network is one type of supervised ANNs that consists of a group of neurons. Furthermore, in MLP, there are three layers. The first one is called the input layer, used to feed the model with input data that passes to the next layer. While, the layers that follow are the middle layer, which is called the hidden layer, and the final layer, which represents the output layer [32].

2.2.3. XGBoost

XGBoost is one of the most popular, strong, and efficient methods for machine learning tasks in the Gradient Boosting Library. It is a combination method that uses gradient boosting based on a decision tree approach. In this algorithm, multiple cores work together to create each tree. Furthermore, the data are formatted to reduce and optimize search times. In addition, this model supports classification, ranking, and regression applications, and is widely used and favored by data scientists because of the flexibility, computational efficiency, and speed [33].

2.2.4. Naïve Bayes

Naive Bayes is a popular statistical and classification supervised method in the ML field based on Bayes Theorem, which assumes attribute independence. It is considered an efficient and fast model with the ability to deal with multiple-dimensional datasets [34]. This method employs the conditional probability principle to compute the similarity of a particular class based on input features [35].

2.2.5. Logistic Regression

Logistic regression is one of the machine learning algorithms, which is widely used for classification tasks and a form of binomial regression. It is used to model the probability of an event whether it will happen or not. There are two possible values in binary classification, typically represented as 0 or 1 and yes or no [36].

2.3. Stacking Ensemble Approach

Stacking is one of the ensemble learning methods that is used to combine multiple ML algorithms to enhance the performance and build more reliable models. This integration of various models helps to reduce misclassification and improve the overall capabilities of the individual models and the system. There are two main core components that this approach is built on, including the base model and the meta-learner. The main idea of combining the two models is to utilize the results of meta learners as the final output [37].

3. Materials and Methods

This section describes the methodology of the proposed system used for the classification task.

3.1. Dataset

A subset of the publicly available dataset on Kaggle [38] for forest fire was adopted to develop and evaluate the proposed model [7]. Consisting of labeled fire images and non-fire images. In this study, a total of 1520 images were selected from the original dataset to reduce computational time while ensuring a representative and balanced sample. All images are RGB with a resolution of 250* 250 pixels. The dataset used in this paper is divided into two categories: fire and non-fire, with 760 images for each class. Figure 1 shows visual sample examples of the fire and non-fire image classes.

Fig. 1- Sample of the dataset.

3.2. Proposed Model

This section describes the main steps of the proposed system employed for the classification task. Figure 2 shows the flowchart of the whole process.

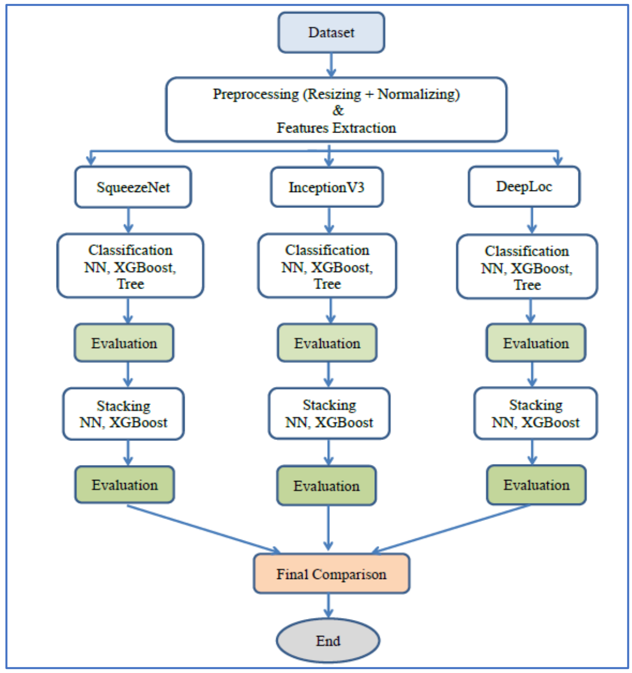

Fig. 2- Flowchart of the proposed method.

3.2.1. Preprocessing and Feature Extraction

The image embedding tool in the Orange platform [29] was used to perform preprocessing of the images, including resizing and normalizing, and feature extraction automatically. This tool uses pre-trained deep learning models to do the task. This step helped save time and contributed to ensuring uniform processes. Additionally, to reduce the errors that might be made through the manual task.

Once the preprocessing task is complete, the preprocessed image is passed through the network model using the same image embedding, and then it is passed through multiple layers, usually convolutional layers. This step is utilized to discriminate and extract high-level features from the dataset of forest fire images for the extraction of feature vectors. Those pre-trained models represented by CNNs, including InceptionV3, SqueezeNet, and DeepLoc, were applied individually to determine which one gives the best performance. The deep learning models selected in this task for several reasons, Inception V3 for computational efficiency and to provide effective features [25], while SqueezeNet is a fast and small model, and DeepLoc for feature extraction capabilities. The feature vectors were extracted implicitly and automatically using the image embedding widget in the Orange Data Mining tool. InceptionV3 used the activation of the penultimate layer as the embedding vector, while SqueezeNet used the outputs of the pre-softmax (flatten10) layer as the image embedding, and DeepLoc used the activations of the penultimate layer(fc_2) for the same reason [29]. The results of pre-trained models will be used as input for the following process, which is the classification step. This approach leverages powerful features extracted by pre-trained CNNs. This will save time without needing to design, develop, and train a new feature extraction model from the beginning.

3.2.2. Classification with ML Models

After the feature extraction process with each of SqueezeNet, DeepLoc, and InceptionV3, various ML models were utilized for the binary classification and assigned each image to the correct class, including Decision Tree (DT), Neural Network (NN), and XGBoost classifiers. In this step, each model was run separately to classify and train the same set of extracted features. This enables us to understand how each method works and ensures a fair comparison for all classifiers. Furthermore, to evaluate and compare their performance using multiple metrics.

Based on this evaluation, the high-performing classifiers are selected for the next step, which is the stacking approach task.

3.2.3. Ensemble using Stacking

For making the classification task more dependable for some cases and to take advantage of the strengths of each classifier, the NN and XGBoost classifiers are combined using a stacking approach. In this step of the proposed system, the output of the models from the base models was passed to another classifier, which is called a meta-learner, for the final decision. Two meta-learners were used, including Logistic Regression and Naive Bayes. Then, they compared their performance with different feature extractors and with individual base classifiers to determine any potential improvement in classification performance. The performance of the stacking task is also evaluated by several metrics. The purpose of the combination is to figure out whether this technique gives better outcomes than the individual models or will fail. In addition, to highlight the importance of choosing the meta classifiers carefully, this affects the accuracy of the outputs significantly. The flowchart of feature extraction and stacking ensemble steps is shown in Figure 3.

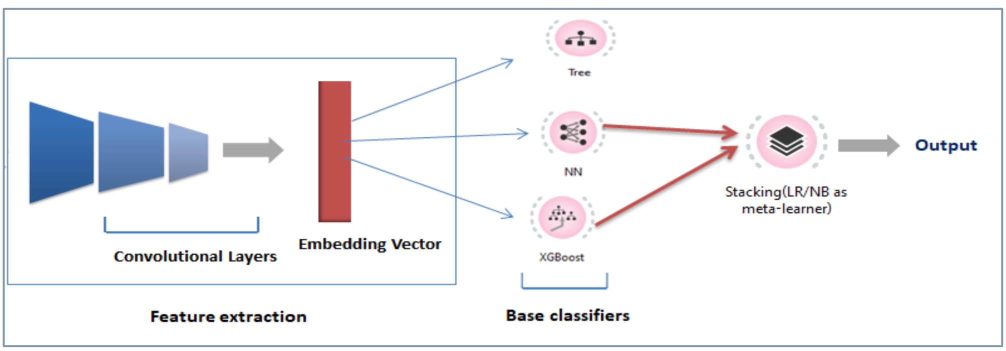

Fig. 3-Flowchart of feature extraction and stacking ensemble steps.

3.3. Evaluation Metrics

Model Performance was evaluated using many standard key metrics, including Accuracy, Precision, F1 score, Recall, True Negative(TN), True Positive(TP), False Positive (FP), and False Negative(FN) [15]. Area Under the Curve (AUC), and Matthews Correlation Coefficient (MCC). A k-fold cross-validation approach with K=5 was employed for better results. This means that the dataset is partitioned into five subsets. Using four parts to train the model, while the remaining one is used for testing in each iteration [39].

The test and score widget in the Orange tool was used for evaluating the model. During cross-validation, the dataset is divided into training and testing sets implicitly for every fold. This cross-validation process was repeated many times, and the results remained consistent across multiple executions. This consistency demonstrates that the results are stable and reliable, gives good confidence, and helps avoid potential overfitting.

4. Results and Discussion

This section examines the effectiveness of various classifiers in detecting and classifying forest fire images. Furthermore, analyze classification results using three pre-trained feature extractors, InceptionV3, SqueezeNet, and DeepLoc, in combination with three ML classifiers, NN, XGBoost, and Decision Tree. The Orange data mining software [28] [29] was used to implement this research paper. The dataset was balanced between the two classes, fire and non-fire images. Furthermore, all experiments were conducted with k-fold cross-validation to guarantee consistency and reliability.

Table 1 presents and provides a comprehensive comparison of models' performance across various image embeddings. As shown in this table, the combination of InceptionV3 with NN achieved the best individual performance and reached a Classification Accuracy (CA) of 98.6 %, an F1-score of 98.6% and MCC of 97.1%. These outcomes highlight the robust ability of InceptionV3 to recognize conventional patterns and effective features due to its deep architecture. While NN, because of its capability of modeling linear and nonlinear features and identifying the complex structure, leads to the ability to classify and distinguish between the fire and non-fire images with high performance. Notably, the Neural Network model reliably achieved better results than both the Decision Tree and XGBoost models. XGBoost achieved 98.3% accuracy, and Decision Tree achieved 95.3% accuracy with InceptionV3. Although SqueezeNet performed well, its performance was slightly lower than that obtained by InceptionV3. For instance, when using with NN, achieved 97.6% of CA, 97.6% of F1-score, and 95.1% of MCC. While XGBoost obtained 96.8% of CA and 93.7% of MCC, and DT reached 92% of CA and 84.1% of MCC.

On the other hand, the DeepLoc model lagged behind. It underperformed across all configurations, particularly when combined with DT, which recorded the weakest overall performance with 80.1% of CA and 60.3% of MCC. This shows the limited ability of this extractor to capture informative features for this classification task.

Table 1: Comparison of performance for classification algorithms on different pre-trained models.

Model	Classifier	AUC	CA	F1-score	Precision	Recall	MCC
InceptionV3	**NN**	**0.999**	**0.986**	**0.986**	**0.986**	**0.986**	**0.971**
InceptionV3	XGBoost	0.999	0.983	0.983	0.983	0.983	0.966
InceptionV3	DT	0.943	0.953	0.953	0.953	0.953	0.907
SqueezeNet	NN	0.997	0.976	0.976	0.976	0.976	0.951
SqueezeNet	XGBoost	0.994	0.968	0.968	0.968	0.968	0.937
SqueezeNet	DT	0.905	0.92	0.92	0.92	0.92	0.841
DeepLoc	NN	0.953	0.885	0.885	0.885	0.885	0.77
DeepLoc	XGBoost	0.946	0.878	0.878	0.878	0.878	0.757
DeepLoc	DT	0.759	0.801	0.801	0.802	0.801	0.603

For further performance enhancement, a stacking approach was applied across all embeddings by combining the top two classifiers (Neural Network and XGBoost). Two different meta-learners are used, including Logistic Regression (LR) and Naive Bayes (NB), to assess their ability to improve the results. In addition, evaluate their impact on the effectiveness of the stacking process.

As shown in Table 2, stacking led to consistent improvements across all embeddings. For example, when applying InceptionV3 with stacking and using Logistic Regression as a meta-learner, it achieved the best classification accuracy of 98.9% and MCC of 97.8% compared to all individual classifiers. Similar gains were observed with SqueezeNet, which obtained 97.8% CA and 89.1% CA with DeepLoc embedder.

While LR improves the performance and steady gains, Naive Bayes performs poorly and drops the results across all embeddings, with accuracy stuck near 50.1%. This shows a clear limitation of NB and struggling with the most high and complex features involved in this process. This leads to the stacking performance being strongly sensitive to the choice of meta-learner and how much it matters.

Table 2: Comparison of stacking results for (NN and XGBoost) using different meta-learners for each embedding.

Model	Meta-learner	AUC	CA	F1-score	Precision	Recall	MCC
InceptionV3	**Logistic Regression**	**0.999**	**0.989**	**0.989**	**0.989**	**0.989**	**0.978**
InceptionV3	Naive Bayes	0.501	0.501	0.496	0.501	0.501	0.001
SqueezeNet	Logistic Regression	0.992	0.978	0.978	0.978	0.978	0.955
SqueezeNet	Naive Bayes	0.5	0.5	0.495	0.5	0.5	0
DeepLoc	Logistic Regression	0.951	0.891	0.891	0.892	0.891	0.783
DeepLoc	Naive Bayes	0.501	0.5	0.333	0.25	0.5	0

To understand and evaluate the benefits of stacking and its contributions to classification outcomes, Table 3 provides a comparison between the two best models based on their two error types: False Negatives (FN) and False Positives (FP). Those are the best base classifier, which is NN with InceptionV3, and the best stack that was introduced by combining NN with XGBoost, using Logistic Regression as a meta-learner over InceptionV3 embedding.

The results show that the stacked model yielded fewer total errors with 10 false negatives and 7 false positives. Compared to 13 false negatives and 9 false positives produced by the NN base model. This reflects a notable decrease in misclassification across both FN and FP errors, and improves the reliability. The reason behind that stacking model outperformed the other individual classifiers because it combines the strengths of those classifiers. This allows the model to capture multiple patterns and improve the overall performance.

Table 3: Comparison of misclassification errors between the best base classifier and the best stacking (NN+ XGBoost) based on FN and FP for each embedding.

Model	Classifier	False Negative(FN)	False Positive (FP)
DeepLoc	NN	93	82
DeepLoc	Stack	89	76
SqueezeNet	NN	19	18
SqueezeNet	Stack	18	16
InceptionV3	NN	13	9
InceptionV3	**Stack**	**10**	**7**

The confusion matrix of the stacking model, which achieved the best model performance, is shown in Figure 4, including the values of FN, FP, TN, and TP. This matrix shows the strong predictive capability of the proposed model. The model successfully identified 750 fire cases, which represent the value of True Positives, and 753 non-fire cases for True Negatives. While only 17 samples were misclassified, with 10 false negatives and 7 false positives. The low rate of errors reflects the high ability of the model to detect fire cases and minimize false alarms with strong precision. Overall, these outcomes highlight that the proposed model attains a high accuracy of 98.9%. Furthermore, it preserves a reliable balance between precision and detection capability. This makes it an effective tool for monitoring and reducing the risks of wildfires.

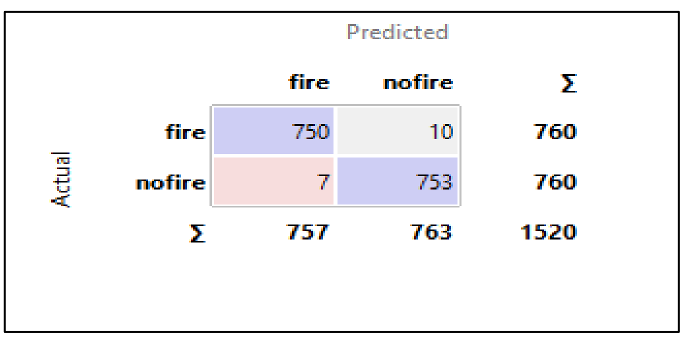

Fig. 4- Confusion matrix of best model (stack using InceptionV3, NN, XGBoost, and LR).

In addition, Figure 5 visually supports these results by presenting a bar chart that compares the error counts between the two models (best stack and best individual classifier) over all embeddings. The chart clearly highlights the advantages of stacking in reducing misclassification, and more effectively than the individual model.

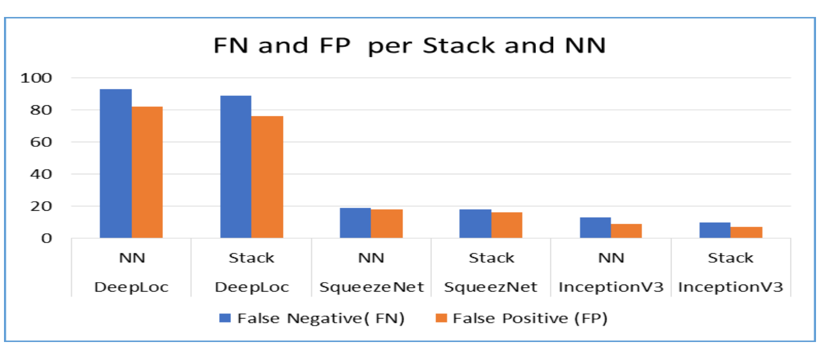

Fig. 5-Errors distribution (FN/FP) for the best base classifier and stacking across pre-trained models.

To highlight the significance and contribution of this work, the proposed work is compared with multiple previous works on the forest fire classification task in Table 4. These studies applied different methods and showed varying outcomes. The table presents the models, reported accuracy, the limitations of these works, and demonstrates improvements and differences of the proposed work.

Table 4: Comparison of proposed work with the related studies.

Reference	Model /Method	Accuracy	Limitation	Comparison to the proposed work
[7]	VGG19	95%	No stacking, focused only on one transfer learning model (VGG19)	Stacking ensemble uses the integration of multiple models to enhance performance by leveraging the strengths of the base models in this work. Comparing many pre-trained models to select the best model.
[8]	ForestResNet based on ResNet50	92%	Limited dataset(173) images , no stacking, focused on the accuracy metric (limited evaluation)	The proposed approach used a larger dataset from a different source, employing a stacking ensemble to enhance robustness. Using several metrics to ensure a comprehensive evaluation. Comparing multiple models to identify the best approach.
[9]	RCNN	97.29%	No stacking, Limited evaluation metrics, based on (accuracy) to evaluate the performance	In the proposed study, enhancing classification performance using stacking ensemble learners and apply various metrics to provide a more comprehensive evaluation for fire classification.
[10]	FFireNet based on MobileNetV2	98.42%	No stacking, the precision with the DeepFire dataset is 97.42%, the recall is 99.47%, and the F1-score is 98.43%	The current work used stacking to improve the performance 98.9% of accuracy, precision, and recall
[11]	EfficientNetB7+ ACNet+Bayesian optimization	95.97% On the DeepFire dataset	No stacking, the precision with DeepFire dataset 95.19%, recall 96..01%, F1-score 95.54%	This work employed stacking and achieved a higher performance with 98.9% of accuracy, precision, recall, and 99.9% AUC.

This work	Best model :InceptionV3+stacking(NN+XGBoost) with LR as meta learner	98.9%		The proposed work evaluated multiple feature extraction models with different classifiers and meta-learners, selecting the best-performing one by applying a stacking ensemble and selecting the best model. This approach enhances and improves the performance of the classification task across multiple metrics.

These improvements are significant in real-world fire detection systems. Minimizing false negative errors helps lower the risk of missing fire events, while reducing the false positives and avoiding unnecessary alerts or responses. Even minor improvements in error rates can make a significant impact in safety-critical applications such as fire detection, where every decision counts.

5. Conclusion

This study demonstrated that integrated deep feature extraction with machine learning classifiers and stacking ensemble has the ability to classify forest fire images and improve the accuracy of this critical task. InceptionV3 attained the highest performance among the tested embeddings, and the Neural Network model obtained better performance compared to XGBoost and Decision Tree. The stacked model of combining NN and XGBoost with Logistic Regression as a meta-learner outperformed all individual models and achieved the highest and best performance.

The choice of meta-learners played a crucial role in the ensemble's overall effectiveness. Logistic Regression led to better performance compared to Naive Bayes in this context. The ability to reduce false negative errors is vital in safety and high-risk applications, especially in fire detection situations. Early accurate detection and classification can significantly affect the outcomes of real-life applications.

While the proposed approach achieved consistent and effective performance in classifying forest fire images, future work may use more diverse datasets and implement it on video sequences rather than images. In addition, exploring more advanced ensemble techniques for real-time systems and real-world environments.

References

[1] L. Zhang, M. Wang, Y. Fu, and Y. Ding, "A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning," *Forests*, vol. 13, no. 7, p. 975, June 2022, doi: 10.3390/f13070975.

[2] D. Han, Y. Guo, J. Wang, and B. Zhao, "Global disparities in indoor wildfire-PM2.5 exposure and mitigation costs," *Sci. Adv.*, vol. 11, no. 20, p. eads4360, May 2025, doi: 10.1126/sciadv.ads4360.

[3] J. Li *et al.*, "The persistence of smoke VOCs indoors: Partitioning, surface cleaning, and air cleaning in a smoke-contaminated house," *Sci. Adv.*, vol. 9, no. 41, p. eadh8263, Oct. 2023, doi: 10.1126/sciadv.adh8263.

[4] S. Moazeni and A. Cerdà, "The impacts of forest fires on watershed hydrological response. A review," *Trees For. People*, vol. 18, p. 100707, Dec. 2024, doi: 10.1016/j.tfp.2024.100707.

[5] T. Khan *et al.*, "Trust-driven approach to enhance early forest fire detection using machine learning," *Sci. Rep.*, vol. 15, no. 1, p. 14480, Apr. 2025, doi: 10.1038/s41598-025-99032-6.

[6] R. N. Vasconcelos *et al.*, "Fire Detection with Deep Learning: A Comprehensive Review," *Land*, vol. 13, no. 10, p. 1696, Oct. 2024, doi: 10.3390/land13101696.

[7] A. Khan, B. Hassan, S. Khan, R. Ahmed, and A. Abuassba, "DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection," *Mob. Inf. Syst.*, vol. 2022, no. 1, p. 5358359, 2022, doi: 10.1155/2022/5358359.

[8] Y. Tang, H. Feng, J. Chen, and Y. Chen, "ForestResNet: A Deep Learning Algorithm for Forest Image Classification," *J. Phys. Conf. Ser.*, vol. 2024, no. 1, p. 012053, Sept. 2021, doi: 10.1088/1742-6596/2024/1/012053.

[9] A. Chopde, A. Magon, and S. Bhatkar, "Forest Fire Detection and Prediction from image processing using RCNN," in *World Congress on Civil, Structural, and Environmental Engineering*, Avestia Publishing, Apr. 2022. doi: 10.11159/iceptp22.204.

[10] S. Khan and A. Khan, "FFireNet: Deep Learning Based Forest Fire Classification and Detection in Smart Cities," *Symmetry*, vol. 14, no. 10, p. 2155, Oct. 2022, doi: 10.3390/sym14102155.

[11] A. M. Islam *et al.*, "An Attention-Guided Deep-Learning-Based Network with Bayesian Optimization for Forest Fire Classification and Localization," *Forests*, vol. 14, no. 10, p. 2080, Oct. 2023, doi: 10.3390/f14102080.

[12] Y. Wang, L. Dang, and J. Ren, "Forest fire image recognition based on convolutional neural network," *J. Algorithms Comput. Technol.*, vol. 13, p. 1748302619887689, Jan. 2019, doi: 10.1177/1748302619887689.

[13] Y. Supriya and T. R. Gadekallu, "Particle Swarm-Based Federated Learning Approach for Early Detection of Forest Fires," *Sustainability*, vol. 15, no. 2, p. 964, Jan. 2023, doi: 10.3390/su15020964.

[14] Q. Xue, H. Lin, and F. Wang, “FCDM: An Improved Forest Fire Classification and Detection Model Based on YOLOv5,” *Forests*, vol. 13, no. 12, p. 2129, Dec. 2022, doi: 10.3390/f13122129.
[15] M. Prakash, S. Neelakandan, M. Tamilselvi, S. Velmurugan, S. Baghavathi Priya, and E. Ofori Martinson, “Deep Learning-Based Wildfire Image Detection and Classification Systems for Controlling Biomass,” *Int. J. Intell. Syst.*, vol. 2023, no. 1, p. 7939516, 2023, doi: 10.1155/2023/7939516.
[16] A. Lertsinsrubtavee, T. Kanabkaew, and S. Raksakietisak, “Detection of forest fires and pollutant plume dispersion using IoT air quality sensors,” *Environ. Pollut.*, vol. 338, p. 122701, Dec. 2023, doi: 10.1016/j.envpol.2023.122701.
[17] M. F. S. Titu, M. A. Pavel, G. K. O. Michael, H. Babar, U. Aman, and R. Khan, “Real-Time Fire Detection: Integrating Lightweight Deep Learning Models on Drones with Edge Computing,” *Drones*, vol. 8, no. 9, p. 483, Sept. 2024, doi: 10.3390/drones8090483.
[18] A. W. Ali and S. Kurnaz, “Optimizing Deep Learning Models for Fire Detection, Classification, and Segmentation Using Satellite Images,” *Fire*, vol. 8, no. 2, p. 36, Jan. 2025, doi: 10.3390/fire8020036.
[19] C. R. Madhuri, S. S. Jandhyala, D. M. Ravuri, and V. D. Babu, “Accurate classification of forest fires in aerial images using ensemble model,” *Bull. Electr. Eng. Inform.*, vol. 13, no. 4, Art. no. 4, Aug. 2024, doi: 10.11591/eei.v13i4.6527.
[20] H. Liang, M. Zhang, and H. Wang, “A Neural Network Model for Wildfire Scale Prediction Using Meteorological Factors,” *IEEE Access*, vol. 7, pp. 176746–176755, 2019, doi: 10.1109/ACCESS.2019.2957837.
[21] Y. Li, G. Li, K. Wang, Z. Wang, and Y. Chen, “Forest Fire Risk Prediction Based on Stacking Ensemble Learning for Yunnan Province of China,” *Fire*, vol. 7, no. 1, p. 13, Dec. 2023, doi: 10.3390/fire7010013.
[22] A. Rathod, V. Pariawala, M. Surana, and K. Saxena, “Leveraging CNNs and Ensemble Learning for Automated Disaster Image Classification,” in *Artificial Intelligence and Sustainable Computing*, M. Pandit, M. K. Gaur, and S. Kumar, Eds., Singapore: Springer Nature, 2024, pp. 311–326. doi: 10.1007/978-981-97-0327-2_23.
[23] P. Doungpaisan and P. Khunarsa, “A comparative study of pre-trained models for image feature extraction in weather image classification using orange data mining,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 37, no. 1, Art. no. 1, Jan. 2025, doi: 10.11591/ijeecs.v37.i1.pp241-249.
[24] M. Mujahid, F. Rustam, R. Álvarez, J. Luis Vidal Mazón, I. de la T. Díez, and I. Ashraf, “Pneumonia Classification from X-ray Images with Inception-V3 and Convolutional Neural Network,” *Diagnostics*, vol. 12, no. 5, p. 1280, May 2022, doi: 10.3390/diagnostics12051280.
[25] K. Srinivas, R. Gagana Sri, K. Pravallika, K. Nishitha, and S. R. Polamuri, “COVID-19 prediction based on hybrid Inception V3 with VGG16 using chest X-ray images,” *Multimed. Tools Appl.*, pp. 1–18, June 2023, doi: 10.1007/s11042-023-15903-y.
[26] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. J. Soufi, “Deep-COVID: Predicting COVID-19 From Chest X-Ray Images Using Deep Transfer Learning,” July 21, 2020, *arXiv*: arXiv:2004.09363. doi: 10.48550/arXiv.2004.09363.
[27] T. Shahriar, “Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices,” May 06, 2025, *arXiv*: arXiv:2505.03303. doi: 10.48550/arXiv.2505.03303.
[28] J. Demšar *et al.*, “Orange: Data Mining Toolbox in Python,” *J. Mach. Learn. Res.*, vol. 14, pp. 2349–2353, 2013.
[29] “Orange Data Mining,” Orange Data Mining. Accessed: Aug. 03, 2025. [Online]. Available: https://orangedatamining.com
[30] A. H. Elsheikh, N. Ma, F. A. Essa, M. Khedr, and A. M. M. Ibrahim, “Kerf geometry prediction and optimization in laser cutting of basalt fiber reinforced polymer composites using decision tree and coati optimization algorithm,” *Results Eng.*, vol. 26, p. 105514, June 2025, doi: 10.1016/j.rineng.2025.105514.
[31] A. Tripathi, T. Goswami, S. K. Trivedi, and R. D. Sharma, “A multi class random forest (MCRF) model for classification of small plant peptides,” *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100029, Nov. 2021, doi: 10.1016/j.jjimei.2021.100029.
[32] J. Su, Z. Lin, F. Xu, G. Fathi, and K. A. Alnowibet, “A hybrid model of ARIMA and MLP with a Grasshopper optimization algorithm for time series forecasting of water quality,” *Sci. Rep.*, vol. 14, no. 1, p. 23927, Oct. 2024, doi: 10.1038/s41598-024-74144-7.
[33] Z. A. Ali, Z. H. Abduljabbar, H. A. Tahir, A. B. Sallow, and S. M. Almufti, “Exploring the Power of eXtreme Gradient Boosting Algorithm in Machine Learning: a Review,” *Acad. J. Nawroz Univ.*, 2023.
[34] O. Peretz, M. Koren, and O. Koren, “Naive Bayes classifier – An ensemble procedure for recall and precision enrichment,” *Eng. Appl. Artif. Intell.*, vol. 136, p. 108972, Oct. 2024, doi: 10.1016/j.engappai.2024.108972.
[35] P. J. B. Pajila, B. G. Sheena, A. Gayathri, J. Aswini, M. Nalini, and S. S. R, “A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications,” in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Sept. 2023, pp. 1228–1234. doi: 10.1109/ICOSEC58147.2023.10276274.
[36] A. Karim, A. Azhari, S. B. Belhaouri, A. A. Qureshi, and M. Ahmad, “Methodology for Analyzing the Traditional Algorithms Performance of User Reviews Using Machine Learning Techniques,” *Algorithms*, vol. 13, no. 8, p. 202, Aug. 2020, doi: 10.3390/a13080202.
[37] M. A. Muslim *et al.*, “An Ensemble Stacking Algorithm to Improve Model Accuracy in Bankruptcy Prediction,” *J. Data Sci. Intell. Syst.*, vol. 2, no. 2, pp. 79–86, Mar. 2023, doi: 10.47852/bonviewJDSIS3202655.
[38] “Forest Fire Dataset.” Accessed: July 26, 2025. [Online]. Available: https://www.kaggle.com/datasets/alik05/forest-fire-dataset
[39] A. V. D. Sano, A. A. Stefanus, E. D. Madyatmadja, H. Nindito, A. Purnomo, and C. P. M. Sianipar, “Proposing a visualized comparative review analysis model on tourism domain using Naïve Bayes classifier,” *Procedia Comput. Sci.*, vol. 227, pp. 482–489, Jan. 2023, doi: 10.1016/j.procs.2023.10.549.