Optimized Weighted Ensemble Voting based Machine Learning Framework for Improved Fire Detection Using Longitudinal Multi Sensor Data

Simhavishnu Ram Prasad[1], Saikat Kumar Shome[2] and Dhiraj Chaurasia[3]

National Institute of Technology Durgapur, M G Avenue, Durgapur-713209 West Bengal India[1] , CSIR-Central Mechanical Engineering Research Institute, M G Avenue, Durgapur – 713209 West Bengal, India[2], National Institute of Technology Durgapur, M G Avenue, Durgapur-713209 West Bengal India [3]

Abstract. Fire outbreaks are serious hazards that can occur anywhere including houses, schools, office buildings and forests which can result in casualties, environmental damage and economic loss. Advanced fire detection systems must be employed to identify and mitigate a fire outbreak in any environment. This paper presents a machine learning and deep learning approach that predicts a result considering inputs from both visual data from surveillance cameras and sensor data. The proposed model uses a novel ensemble learning algorithm to combine convolutional neural networks and random forest regression to predict a fire outbreak more efficiently and reduce false alarms. The results show that the proposed framework was efficiently able to distinguish between fire and non-fire with very high accuracy and low false positives. The results can be improved by further training.

Keywords:
Fire outbreaks, detection, visual data, sensor data, ensemble learning, machine learning

1. Introduction

According to India Risk Survey (IRS) 2018, fire outbreaks are one of the biggest risks to businesses and operations. Fire outbreaks were ranked the eighth biggest risk to business in IRS 2016 [1]. According to the Global Forest Watch, India recorded 82,170 forest fire alerts in 2021, nearly double what was recorded last year. According to the data, the number of forest fire alerts in April 2021 is also the highest in five years [2]. In addition to the millions of acres of land being destroyed every year, fires release a lethal amount of carbon dioxide and smoke into the atmosphere. A study from the National Science Foundation estimated that fires in the United States release approximately 290 million metric tons of carbon dioxide annually [3]. According to the Global Fire Emissions Database, the carbon emissions from fire outbreaks globally increased by 26 percent to 7.8 billion metric tons in 2020, the highest amount recorded since 2002 [4].

Fire outbreaks can occur unexpectedly because of different reasons. They can be caused naturally due to lightning and low humidity or as a consequence of erratic human activities. Robust and accurate fire detection systems must be employed to prevent such unprecedented hazards and suppress the threat as soon as it arises. The ever-expanding and rapidly growing technology in the world has brought rise to many new intelligent fire detection methods. Nonetheless, many problems remain unsolved, and systems are not in place to take care of unforeseen circumstances. According to the US Drought Monitor, 95 percent of western states in the United States were experiencing significant droughts, and wildfire warnings are set to appear a month ahead of schedule in late 2021[5].

Usually, fire detection systems employ either sensor-based methods or computer vision techniques to detect an outbreak. Most fire alarm systems traditionally use an array of sensor modules such as flame, smoke, and temperature detectors to predict an outbreak. Jadon et al. proposed a smoke sensor-based machine learning and IoT model for early detection of fire [6]. Mahmud et al. proposed an integrated sensor system (ISS) model where smoke, heat, and flame detection sensors are used along with a camera to create an early fire detection framework [7].

A delay in detection time can lead to unwanted and unnecessary damage. In addition, false alarms can be expensive and lead to wasteful use of resources. In 2018, the U.S Fire Department reported a total of 2,889,000 false alarms, a lot of which were caused due to system malfunctions [8]. To overcome such shortcomings posed by sensor-based fire detection systems, researchers have developed computer vision methods to detect fire outbreaks. Kim et al. proposed an intelligent RGB classifier that uses movement detection, color analysis, and blob analysis for fire detection [9]. Valikhujaev et al. proposed a convolutional neural network approach that uses dilated convolutions to provide accurate predictions and overcome challenges posed by traditional fire alarms [10].

Another common technique many researchers use is the contrast between the candidate fire region and the environment to detect fire outbreaks effectively. Mueller et al. proposed a method that uses the idea of differentiating between the fast, turbulent movement of flame and the firm and consistent motion of other objects. [11] Similar to this method, Giuseppe et al. proposed a fire detection framework to filter out a candidate fire region and extract dominant fire characteristics from the chosen region [12].

Sensor-based frameworks have many limitations. Sensors require a particular threshold of intensity for an alarm to be triggered, resulting in a delay in detection time. Furthermore, sensors are often unable to distinguish between fire and smoke, which can trigger false alarms. Vision-based fire detection models overcome many limitations posed by sensor-based models, but these models have shortcomings of their own. It is difficult and uneconomical for large areas such as forests to place enough surveillance cameras to encompass the entire area and monitor each crevasse. Vision-based models struggle to predict the extent of a fire and are also vulnerable to variable environmental conditions such as luminescence, perspective distortion, and obstruction. This paper presents a novel ensemble weighted voting method which integrates a convolutional neural network with a random forest classifier to fulfill the above-mentioned loopholes of traditional sensor-based and visual classifiers.

2. Proposed Work

2.1 *Algorithm Proposition*

This paper focuses on intelligent fire detection using a weighted ensemble voting technique. Our proposed model uses two different machine learning and deep learning classifiers that were trained separately to later produce a consolidated prediction on whether there is a fire outbreak or not. The first classifier was a MobileNet based convolutional neural network which was trained on a dataset of fire images. The second classifier was a machine learning classifier that used the random forest algorithm to generate an output. The random forest classifier was trained on sensor data from an experiment that simulated real life fires in different scenarios. A detailed description and analysis of each part is discussed in the following subheadings.

2.1.1 Ensemble Learning

Ensemble learning is a method used in statistics and machine learning where a confluence of multiple individually trained classifiers is used [13]. Combining the predictions from the classifiers can often result in better-performing models than predictions from individual classifiers [14]. Ensemble voting techniques have proven to be highly effective in the past and have displayed better prediction performance than other techniques [15]. Zhou et al. [16] proposed a democratic co-learning method by combining predictions from multiple classifiers using majority voting. The algorithm outperformed individual classifiers proving that combining predictions can be an effective technique. Livieris [17] recently developed a new ensemble technique that used voting based on maximum probability. The model combined predictions from three different learners (co-training, tri-training, and self-training), and the results outperformed those of the classifiers individually.

2.1.2 Ensemble Voting in a Neighborhood

The model uses a dedicated camera and sensor nodes positioned around a given area strategically to create an optimized fire detection system. In areas such as a forest, surveillance cameras can be installed around the periphery of the area to monitor larger fires or fires close to the exterior of the region. In addition, an array of sensor modules can be placed along the intricacies of the target area to detect fire-related indications. Each node will send real-time reports to their dedicated CNN and random forest classifiers to return predictions which shall further be processed using the ensemble voting method to generate the final result. A weighted ensemble method was used, and the weights of each node were assigned according to the proximity of the nodes to the reporting node. The proximity of each node to one another was represented by the Euclidean distance between the nodes. Figure 1 shows the setup of the proposed method.

Let us consider our network as a grid of length n and breadth m. If a node N_{xy} reports a fire outbreak, we perform ensemble voting in a closed neighborhood R (of radius r units of the grid). Radius r is experimentally determined based on crowding and sensitivity.

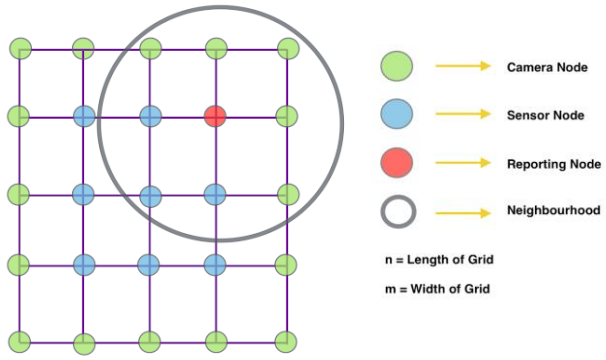


Figure 1: Framework for the Setup of the Proposed Ensemble Method

Let us consider,

$$Reporting\ Node\ =\ Hot\ Node$$
$$Neighbouring\ nodes\ to\ the\ hot\ nodes\ =\ Warm\ Nodes$$
$$Nodes\ out\ of\ the\ neighbourhood\ =\ Cold\ Nodes$$

We assign weights to the nodes based on the heuristic that nodes in close vicinity of fire report more accurately. Hence, the Hot Node gets the most weightage while the weightage of neighboring Warm Nodes is assigned in proportion to the Euclidean distance from the Hot Node. If multiple nodes report an outbreak, we select the node producing the highest probability as the reporting node. Then,

$$O_{xy} = \frac{1}{\alpha\beta}\sum_{i=X}^{X'}\sum_{j=Y}^{Y'}\lambda_{ij}P_{ij} \qquad (1)$$

$$\lambda_{ij} = \frac{1}{e^{D_{ij}}} \quad (2)$$

$$D_{ij} = \sqrt{(x-i)^2 + (y-j)^2} \quad (3)$$

where O_{xy} is the ensemble probability of fire outbreak based on the report of Hot Node N_{xy}, P_{ij} is the probability of fire outbreak reported by the node N_{ij}, λ_{ij} is the weight of Warm Node calculated based on Euclidean distance D_{ij} from the Hot Node. If (x, y) is the position of the Hot Node N_{xy} in the grid and r is the radius of the neighborhood as defined, then

$$X = \max(x - r, 1) \quad (4)$$

$$Y = \max(y - r, 1) \quad (5)$$

$$X' = \min(x + r, n) \quad (6)$$

$$Y' = \min(y + r, m) \quad (7)$$

$$\alpha = X' - X + 1 \quad (8)$$

$$\beta = Y' - Y + 1 \quad (9)$$

where X, X', Y, and Y' helps define the boundary of the neighborhood to search, whereas α and β are used for normalization.

Table 1: Comparison of Ensemble Methods based on Mean Squared Error (MSE) and Mean Absolute Error (MAE)

Validation				
	Weighted Voting	Hybrid Ensemble	Weighted Averaging	RCGA Ensemble
MSE	0.003	0.0033	0.0036	0.0035
MAE	0.0248	0.0286	0.0333	0.0482
Testing				
	Weighted Voting	Hybrid Ensemble	Weighted Averaging	RCGA Ensemble
MSE	0.0028	0.0031	0.0036	0.0033
MAE	0.0266	0.0284	0.0363	0.0484

To choose the best ensemble method suitable for this experiment, four different ensemble techniques: weighted ensemble voting, weighted ensemble averaging, ensemble hybrid and ensemble RCGA-FCM were compared. To obtain the classification errors, the validation and testing procedures were implemented individually on the datasets, and the mean squared errors and mean absolute errors were calculated. Table 1 shows the MSE and MAE values of each method. From the above table, it is clear that the weighted voting method outperformed other ensemble methods in both instances of validation and testing.

2.2 *Framework*

We propose a mesh-based framework, as illustrated in Figure 1. The outer green nodes are RGB camera-based visual classifiers, and the inner blue nodes are sensor fusion-based classifier nodes. The schematic of the framework is depicted in Figure 2. The details of the nodes are discussed in the following subheadings.

2.2.1 Visual Classifier

The visual classifier, at its core, is a CNN deployed on an embedded system. Convolutional neural networks (CNNs) are frameworks used in deep learning and computer vision to analyze visual imagery whose architecture is parallel to the visual cortex of the human brain [18]. CNNs are supervised learning frameworks that are primarily used in image and pattern recognition domains. They take raw image vectors as input and run it through a series of layers containing interconnected nodes referred to as neurons to classify data [19].

CNNs consist of three essential layers: the convolutional layers, pooling layers, and the fully connected layer. CNNs take images as input, and images come in several varied colour spaces such as RGB, HSV, greyscale, etc. [20]. Since computation can often get intensive once images reach higher dimensionalities, a convolutional layer is needed. Here, elements responsible for executing the convolution operation known as kernels or filters are present. The kernel is a sub-matrix in the image that traverses the entire image according to the selected stride value. Through this, prominent features can be extracted from the image, and feature maps are generated.

The second layer, known as the pooling layer is similar to the convolution layer as it is responsible for dimensionality reduction. In this layer, several different activation methods can be used to extract dominant features from a small neighborhood. Max pooling and average pooling are examples of pooling methods.

The final layer, known as the fully connected layer, is the layer that brings everything together. After the input image has been flattened into a column vector, the feed-forward neural network known as the fully connected layer trains the model over a series of epochs and, through distinguishing certain features, is able to classify the images. The final prediction is then later passed on to the output layer.

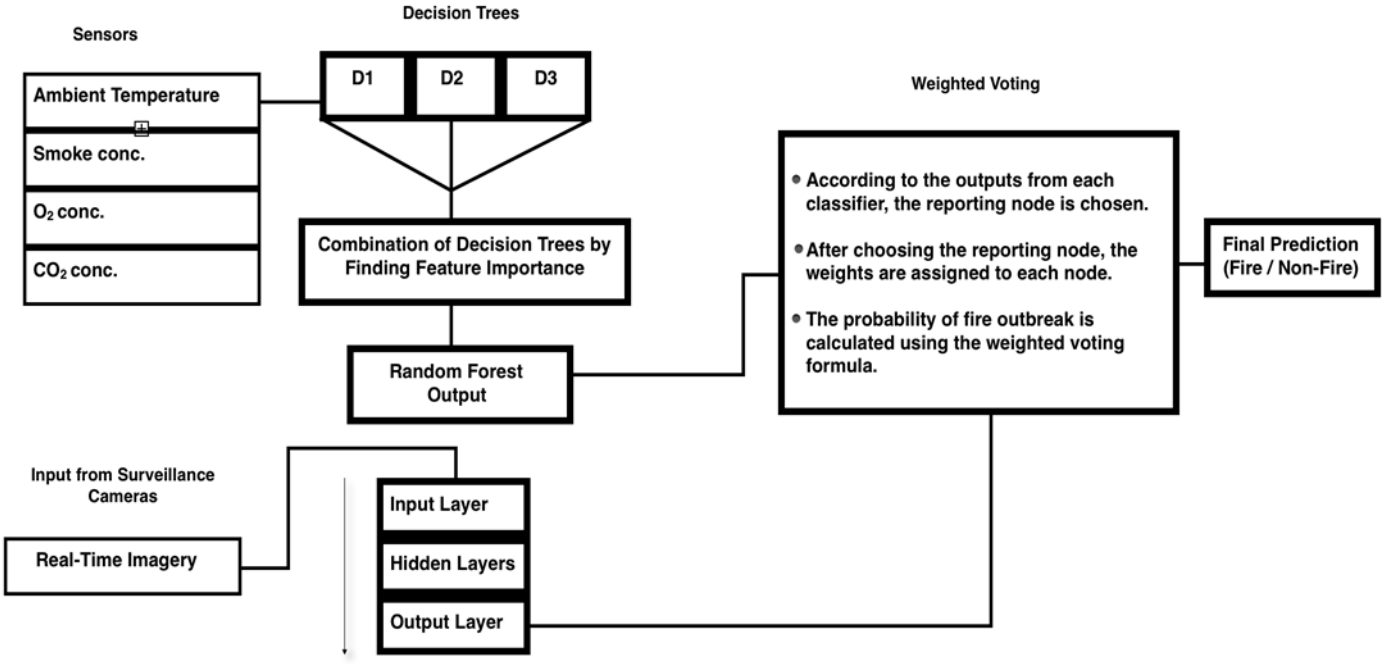


Figure 2: Schematic of the proposed framework

The architecture of the CNN used was a variation of the MobileNet framework as shown in Figure 3. Changes were made to adapt to the problem better. The proposed architecture consists of an input layer, 13 convolutional layers, 12 depth-wise convolutional layers, one average pooling layer followed by a fully connected layer. The size of the images fed into the neural network is 224 x 224 x 3 pixels. To generate feature maps, kernels of size 3 x 3 x 3 x 32 are positioned with a stride length of 2. In the next layer, which is the depth-wise convolutional layer, kernels of size 3 x 3 x 32 are applied with a stride length of 1. These two layers were alternated several times till a 7 x 7 average pooling layer was added of stride length 1. After this, a fully connected layer whose size was 1024 x 1000 was added. Finally, the softmax classifier was added to produce the final prediction.

The key concept behind the MobileNet framework is to use depth-wise separable 3x3 convolutional filters preceding 1x1 kernels rather than the regular 3x3 convolutional filters [21][31]. MobileNet can provide the same combination process and filtering with improved computational efficiency as it requires fewer parameters and operations [22].

Table 2: Comparison of CNN Architectures

Model	No. of Parameters	Feature Size
MobileNet	3.2 M	1024
Resnet	23.5 M	2048
DenseNet	7 M	1024
VGG-19	140 M	4096

Table 2 compares the MobileNet architecture with other commonly used CNN architectures. The ResNet, DenseNet, and VGG-19 are popular architectures that have proven to be highly effective in the past. However, the number of parameters and the feature size of these frameworks are quite large, making them very computationally intensive. So, the MobileNet architecture proves to be the least complex while still providing high performance and accurate results.

2.2.2 Fusion Based Classifier

The inner nodes of our mesh, i.e., the fusion-based classifier, at its core, are Random Forest classifier models deployed on arrays of ambient sensors. Random Forest is a supervised machine learning algorithm where aggregation of decision trees is employed. It is an ensemble learning technique commonly used for classification and regression. Random Forest models are usually trained using the "bagging" method, where the model is trained on data from various subsets of the training set. To calculate the feature importance of the nodes of each decision tree, the Gini Importance is used. The Gini Importance decides the probability of a particular decision tree branch to occur. Let us take pi_j to be the importance of node j. Then,

$$pi_j = w_j c_j - w_{left(j)} c_{left(j)} - w_{right(j)} - c_{right(j)} \quad (10)$$

where w_j is the weight distribution for node j, c_j is the impurity of node j, $left(j)$ is the left child node from node j, $right(j)$ is the right child node from node j. After the Gini Importance, the feature importance values of a decision tree are calculated as

$$fi_i = \frac{\sum_{j:node\ j\ solits\ on\ feature\ i} pi_j}{\sum_{k\ \epsilon\ all\ nodes} pi_k} \quad (11)$$

where fi_i is the feature importance of i, and pi_j is the importance of node j. Then, we further normalize the value between 0 and 1 as

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j} \tag{12}$$

Once we have the normalized feature importance of all the trees, the final feature importance is calculated as the average over all the trees

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T} \tag{13}$$

The final compiled dataset consisted of 1900 images with 950 images belonging to each class (fire and non-fire). The images were reshaped to input size of 224 x 224 for it to be accepted by the CNN. The dataset was split into three different sets for training, testing, and validation.

The model showed a training loss of 0.1509, a validation loss of 0.0486, and a validation accuracy of 0.9819. The training loss is a metric calculated by summing each of them during the training period. The validation loss is a good indicator of how well the model fits new data. When the training loss is less than the validation loss, we assume the model has overfitted to the dataset. In this CNN, the training loss is slightly higher than the validation loss, which means the model has not been overexposed to the training data. In an ideal situation, a perfectly fitted model is when the training loss equals the validation loss.

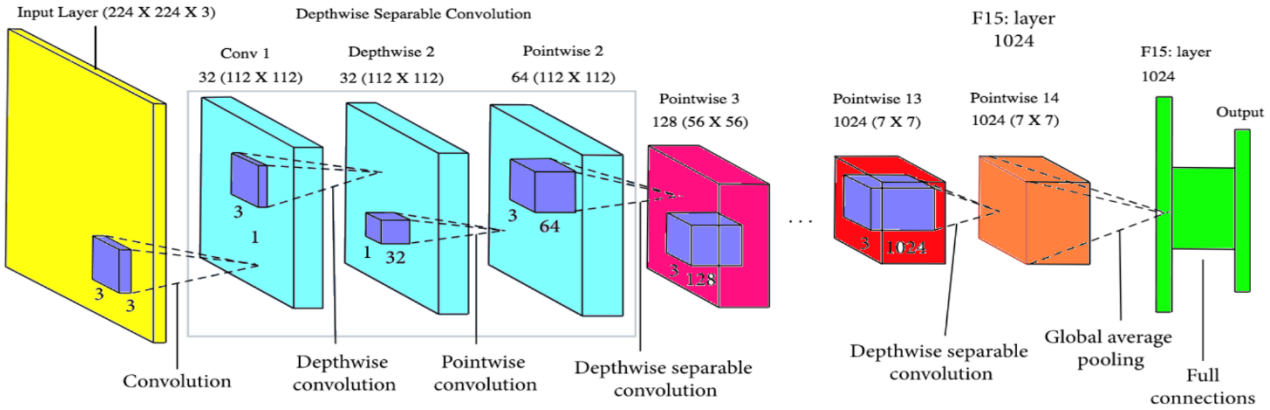


Figure 3: Architecture of Visual Classifier

where T denotes the number of trees.

The fusion-based classifier used in this paper is a modification of our previous work [23]. An array of five ambient sensors: temperature, smoke obscuration, carbon dioxide concentration, and oxygen concentration were used. Values obtained from these sensors were later fed into a microcontroller connected to the sensors through I/O pins. The data is then sent to the data receptor through a Wi-Fi module, where the data is pre-processed and fed into the random forest classifier to predict a real-time result.

2.3 *Training and Dataset*

Both the CNN and the random forest models were trained on a system having the specifications of an Intel(R) Core (T.M.) i7– 10510U core processor @ 1.80GHz and 2666 MHz RAM 16 G.B. and a 1.87Ghz 8GB NVIDIA GeForce 3060 Ti graphics card. The details of the training are discussed in the following subsections.

2.3.1 Training the Visual Classifier

The CNN was trained on a manually compiled dataset containing sensor values and fire images extracted from three different sources. The first source was an experiment by Grammalidis et al. [24], where a multi-sensor fire detection network was built for the protection of cultural heritage sites. The second dataset was a dataset compiled by Khan et al. [25] to detect forest fires. The third dataset used was the FLAME (Fire Luminosity Airborne-based Machine learning Evaluation) dataset. The dataset is comprised of a series of aerial fire images captured using infrared cameras and thermal heat maps. They were used to aid researchers and firefighters to mitigate fire outbreaks [26].

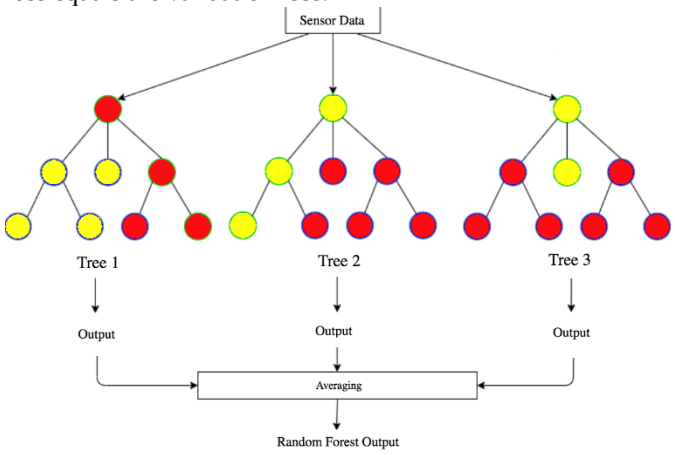


Figure 4:Architecture of Fusion Based Classifier

Fig 5 is an example of a learning curve demonstrating an underfit model. An underfit model can be identified by a continuously decreasing training loss. Fig 6 shows an overfitted model. An overfit model can be confirmed if the validation loss starts to increase after a continuous decrease up to a certain point. Another way of identifying an overfit model is when the training loss decreases as the model becomes more experienced. Fig 7 is an example of a well-fitted model. A well-fitted model can be confirmed if the training and validation loss both decrease to the point of stability with a minute gap between both the plots.

Considering the nature of the dataset, the CNN was trained over 3 epochs with 140 validation steps and 340 steps per epoch to strike a balance and avoid underfitting and overfitting.

2.3.2 Training the Fusion Based Classifier

The random forest classification model was trained on a dataset obtained from an experiment conducted by NIST [23]. Hazardous fire situations were recreated in a controlled environment and CO_2 and O_2 concentrations were recorded along with the smoke obscuration and ambient temperature. An aggregation of 7 datasets gave a total entry amount of 5450 individual values over which the model was trained and tested. These values were passed to the classifier in the form of vectors, where, each vector contained an ordered set of sensor inputs. The dataset was split so that 80% was to be used as the training set and 20% was to be used as the test set.

3. RESULTS

3.1 *The Metrics*

The proposed model was evaluated over several different performance metrics, which include the accuracy score, precision, recall, F1 score, AUC-ROC (Area Under the Curve - Receiver Operating Characteristic) curve, and logarithmic loss.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

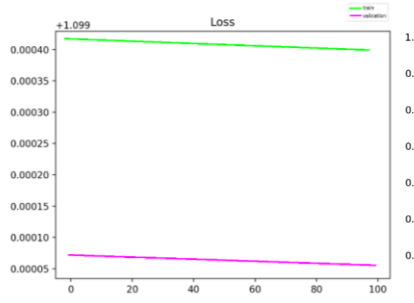


Figure 3: Underfitting

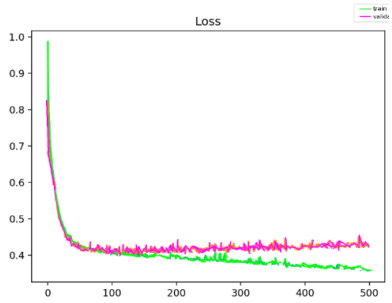


Figure 4: Overfitting

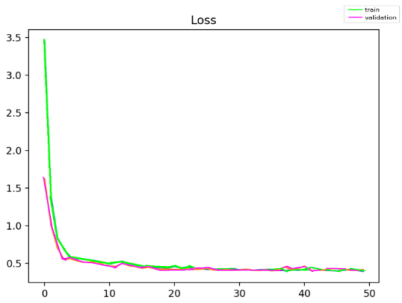


Figure 5: Well-fitted

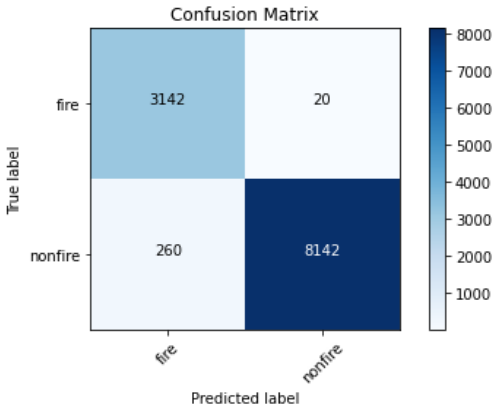


Figure 6: Confusion Matrix of Visual Classifier

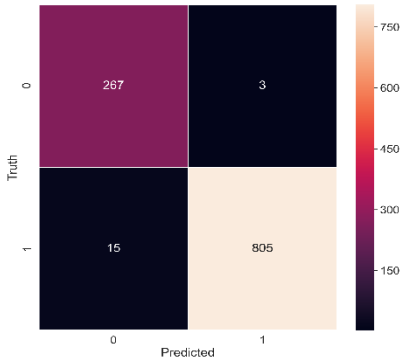


Figure 7: Confusion Matrix of Fusion Based Classifier

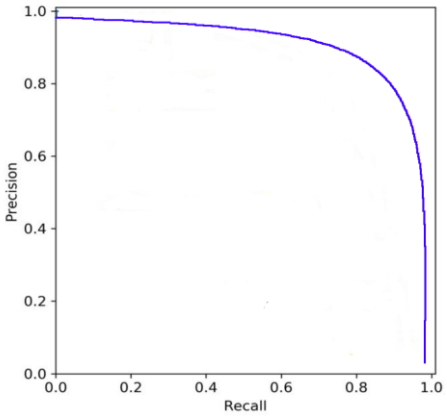


Figure 8: Precision Recall Curve (Visual

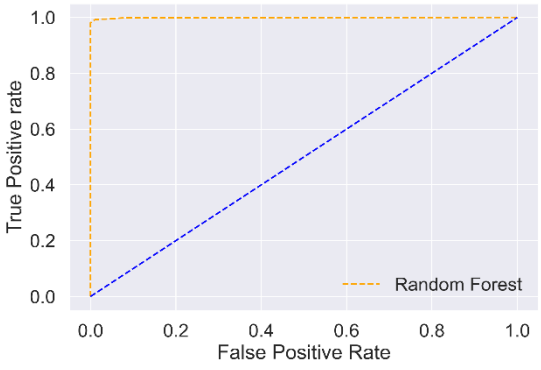


Figure 9: AUC-ROC Curve (Fusion Based Classifier)

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{17}$$

where T.P. is True Positives, TN is True Negatives, F.P. is False Positives and F.N. is False Negatives.

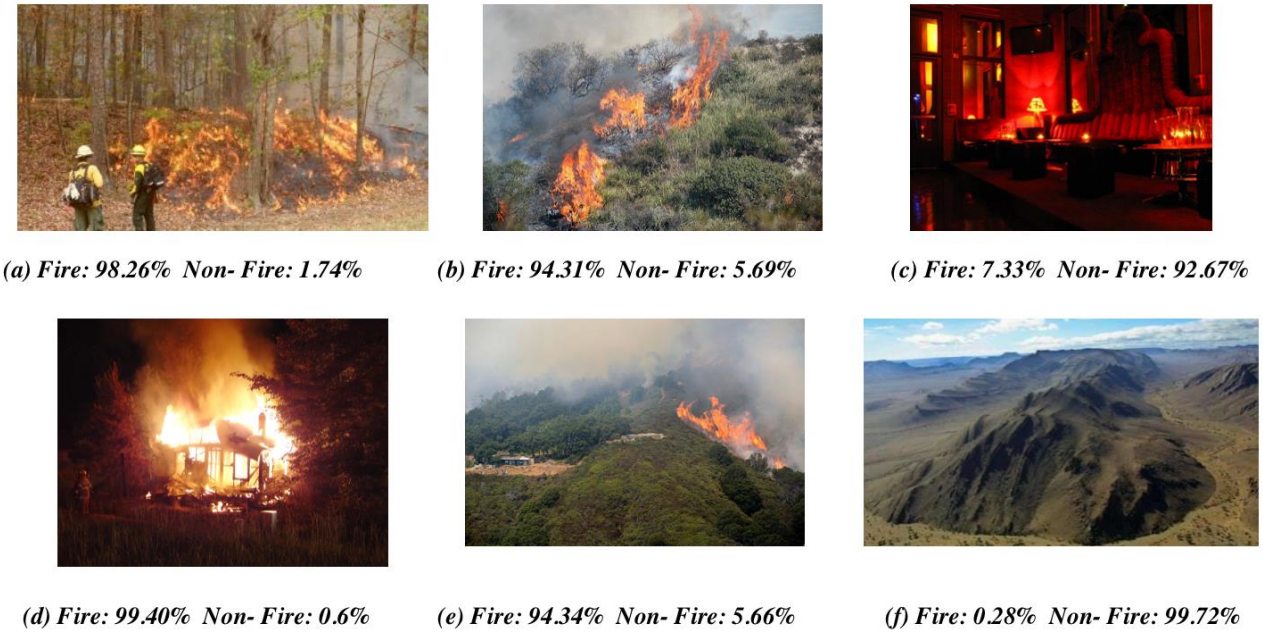
$$Log\ loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij}) \tag{13}$$

where y_{ij} indicates whether sample i belongs to class j or not and p_{ij} indicates the probability of sample i belonging to class j.

rates. The recall score was 0.98. The recall is the ratio of the correctly predicted positive values to all the observations in a class. Just as precision is a good metric for evaluating false positives, recall is suitable for assessing false negatives, which is crucial in threatening cases such as fires. Finally, the F1 score of the model was 0.97. The F1 score is used to represent the harmonic mean between the precision and recall. The F1 score is a handy metric as it takes both false positives and false negatives into account.

Figure 8 shows the precision-recall curve of the CNN. A precision-recall curve was plotted to help visualize the tradeoff between the precision and recall values for different classification thresholds.

This method was compared to other existing methods: Sinha et al. [28], Howard et al. [29], Thomas et al. [30], and Jadon et al. [6], as shown in Table 3, where the precision, recall, and F1-scores of each model was considered. It can be observed that our visual classifier outperforms existing state-of-the-art models.



(a) Fire: 98.26% Non- Fire: 1.74% *(b) Fire: 94.31% Non- Fire: 5.69%* *(c) Fire: 7.33% Non- Fire: 92.67%*

(d) Fire: 99.40% Non- Fire: 0.6% *(e) Fire: 94.34% Non- Fire: 5.66%* *(f) Fire: 0.28% Non- Fire: 99.72%*

Figure 10: Sample classification from the test dataset

3.2 *Performance of the Visual Classifier*

The CNN yielded an accuracy score of 0.9758, indicating that it could classify between fire and non-fire with a high level of validity. The CNN yielded an accuracy score of 0.9758 suggesting that it could classify between fire and non-fire with a high level of validity.

As indicated by the confusion matrix in Figure 6, the CNN reported 8142 true positives, 3142 false negatives, 260 true negatives, and 20 false positives. The model produced a high precision score of 0.97. Precision is the ratio of the correctly predicted positive values to the total positive predicted values. A high precision score relates to lower false positive rates. In this case, precision was picked as a performance metric to evaluate the model as it indicates the extent of false alarm

Table 3: Comparison with existing methods

	Our Model	Sinha et al. [28]	Howard et al. [29]	Thomas et al. [30]	Jadon et al. [6]
Precision	0.97	0.82	0.4 - 0.6	0.6 - 0.7	0.97
Recall	0.98	0.98	0.6 - 0.8	0.4 - 0.5	0.94
F1-Score	0.97	0.89	0.6 - 0.7	0.5 - 0.6	0.95

Figure 10 shows the probabilities that the classifier produced when run over a few random samples from the test set of the dataset. It can be observed that the model performed reasonably well.

3.3 *Performance of the Fusion Based Classifier*

The classifier was evaluated on a set of carefully selected performance metrics: the accuracy score, the AUC-ROC (Area Under the Curve - Receiver Operating Characteristic) curve, and the logarithmic loss. The accuracy score is measured by calculating the ratio of the total number of correct predictions to the total number of predictions. It gives an outlook on how precise the model is. The AUC-ROC curve is a metric to evaluate binary classification models. An AUC score closer to 1 indicates a good measure of separability. An AUC of 0.5 means that the model is incapable of separating two classes. The logarithmic loss is a metric used to evaluate models in predictive modeling problems. It takes into account the model probabilities and is a good measure to assess the model's accuracy. A logarithmic loss score closer to 0 is an indicator of higher accuracy.

The random forest classifier predicted 805 true positives, 267 false negatives, 15 true negatives, and 3 false positives, yielding a final accuracy score of 0.98990. Thus, the model was efficiently able to distinguish fire and non-fire attributes from the given sensor data.

As seen from Figure 9, the classifier produced an AUC score of 0.9974 The AUC score of the model is very high, which indicates excellent performance against all the possible classification thresholds. A logarithmic loss of 0.3398 was recorded. The low logarithmic loss score shows that the model's predictions deviated very little from the actual values. It can thus be inferred that the triggering of false alarms will also be minimal.

4. Conclusion

In this paper, a novel ensemble voting method and architecture for fire detection was proposed. The method involved integrating a sensor-based random forest classifier with a visual convolutional neural network. A setup for the ensemble technique was also displayed, along with a comparison of the method with other ensemble frameworks. To create the most optimized classifiers, both the sensor-based model and the CNN were compared to other techniques and algorithms. After performing a cross-validation test, the random forest classifier reported the least error rate. The MobileNet architecture was used to construct the CNN. This framework was compared to other architectures and proved to be the most efficient after it was compared using different attributes. Finally, the classifiers were evaluated by attentively selected performance metrics. The random forest model reported an accuracy score of 0.98990, and the CNN reported a score of 0.9758. Both the classifiers performed very well across all the metrics and proved to differentiate between fire and non-fire successfully.

Acknowledgment

Authors acknowledge the grant from SEED division of Department of Science and Technology, Govt of India for funding the research. Continuous support of Director, CMERI is also gratefully acknowledged for executing this project.

References

[1] G. Kharmalki et al., "India Risk Survey 2018", Ficci.in, 2021.[Online].Available:https://ficci.in/Sedocument/20450/India%20Risk%20Survey%20-%202018.pdf. [Accessed: 18- Jul- 2021].

[2] Palicha, D., 2021. Forest fires in India: Alerts since April 1 nearly double that of 2020. [online] Downtoearth.org.in.Availableat:<https://www.downtoearth.org.in/news/forests/forest-fires-in-india-alerts-since-april-1-nearly-double-that-of-2020-76559> [Accessed 18 July 2021].

[3] Nsf.gov. 2021. U.S. Fires Release Enormous Amounts of Carbon Dioxide. [online] Available at: <https://www.nsf.gov/news/[Accessed 30 July 2021].

[4] Millan Lombrana, L., Warren, H. and Rathi, A., 2021. Measuring the Carbon-Dioxide Cost of Last Year's Worldwide Wildfires. Bloomberg.

[5] Hartman, A., 2021. US Drought Monitor West. [online] Droughtmonitor.unl.edu. Available at: <https://droughtmonitor.unl.edu/data/png/20210713/20210713_west_trd.png> [Accessed 30 July 2021].

[6] Jadon, Arpit & Omama, Mohd & Varshney, Akshay & Ansari, Samar & Sharma, Rishabh. (2019). FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications.

[7] Mahmud, Mohammad & Islam, Md & Rahman, Md. (2017). Smart Fire Detection System with Early Notifications Using Machine Learning. International Journal of Computational Intelligence and Applications.16.1750009. 10.1142/S1469026817500092.

[8] "U.S. fire statistics: number of false alarms, by type 2018 | Statista", Statista, 2021. [Online]. Available: https://www.statista.com/statistics/376692/number-of-false-fire-alarms-in-the-us-by-type/. [Accessed: 18- Jul- 2021].

[9] Kim, Yoon-Ho & Kim, Alla & Jeong, Hwa-Young. (2014). RGB Color Model Based the Fire Detection Algorithm in Video Sequences on Wireless Sensor Network. International Journal of Distributed Sensor Networks. 2014. 1-10. 10.1155/2014/923609.

[10] Valikhujaev, Yakhyokhuja, Akmalbek Abdusalomov, and Young I. Cho 2020. "Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs" Atmosphere 11, no. 11: 1241. https://doi.org/10.3390/atmos11111241

[11] Mueller, Martin & Karasev, Peter & Kolesov, Ivan & Tannenbaum, Allen. (2013). Optical Flow Estimation for Flame Detection in Videos. IEEE transactions on image

processing : a publication of the IEEE Signal Processing Society. 22. 10.1109/TIP.2013.2258353.

[12] Giuseppe Marbach, Markus Loepfe, Thomas Brupbacher." An image processing technique for fire detection in video images",Fire Safety Journal, Volume 41, Issue 4, 2006, Pages 285-289,ISSN0379-7112 https://doi.org/10.1016/j.firesaf.2006.02.001.

[13] Richard D. P., Jason D. A., Richard W. B., and Paul A. R.: Home Smoke Alarm Project, Manufactured Home Tests at Building and Fire Research Laboratory. National Institute of Standards and Technology. NIST Report of Test FR 4016, (2005).

[14] Valentini, Giorgio & Masulli, Francesco. (2002). Ensembles of Learning Machines. Neural Nets WIRN Vietri-2002, Series Lecture Notes in Computer Sciences. 2486. 3-22. 10.1007/3-540-45808-5_1.

[15] H. Li, et al. Ensemble learning for overall power conversion efficiency of the all-organic dye-sensitized solar cells IEEE Access, 6 (2018), pp. 34118-34126, 10.1109/ACCESS.2018.285004813.

[16] Zhou, Y.; Goldman, S. Democratic co-learning. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Boca Raton, FL, USA, 15–17 November 2014; IEEE: Piscataway, N.I., USA, 2004; pp. 594–602.

[17] Pintelas, P. & Livieris, Ioannis. (2020). Ensemble learning and their applications.

[18] Khan, Ali; Hassan, Bilal (2020), "Dataset for Forest Fire Detection", Mendeley Data, V1, doi: 10.17632/gjmr63rz2r.1

[19] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology(ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[20] D. Chaurasia, S. Shome and P. Bhattacharjee, "Machine Learning Based Intelligent Fire Outbreak Detection in Wireless Sensor Networks", Springer, 2021.]Available:https://www.springerprofessional.de/en/intelligent-fire-outbreak-detection-in-wireless-sensor-networks/19096188. [Accessed: 18- Jul- 2021].

[21] Muhammad, Khan & Ahmad, Jamil & Baik, Sung. (2017). Early Fire Detection using Convolutional Neural Networks during Surveillance for Effective Disaster-Management-Neurocomputing. 10.1016/j.neucom.2017.04.083.

[22] D. Y. T. Chino, L. P. S. Avalhais, J. F. Rodrigues and A. J. M. Traina, "BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis," 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, 2015, pp. 95-102, doi: 10.1109/SIBGRAPI.2015.19.

[23] Rudz, K. Chetehouna, A. Hafiane, H. Laurent, and O. Séro-Guillaume, "nvestigation of a novel image segmentation method dedicated to forestfire applications,''Meas. Sci. Technol., vol. 24, no. 7, p. 075403, 2013.

[24] Visa, Sofia & Ramsay, Brian & Ralescu, Anca & Knaap, Esther. (2011). Confusion Matrix-based Feature Selection.. CEUR Workshop Proceedings. 710. 120-127.

[25] O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.

[26] Zheng Yi 2018 J. Phys.: Conf. Ser. 1087 062018

[27] Khasoggi, Barlian & Ermatita, Ermatita & Samsuryadi, Samsuryadi. (2019). Efficient mobilenet architecture as image recognition on mobile and embedded devices. Indonesian Journal of Electrical Engineering and-Computer-Science.16.389. 10.11591/ijeecs.v16.i1.pp389-394.

[28] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 02800285,doi:10.1109/UEMCON47517.2019.8993089.

[29] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[30] Thomas, C., 2021. An introduction to Convolutional Neural Networks. [online] Medium. Available at: <https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7> [Accessed 30 July 2021].