International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0137

e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 845-848

Wildfire Monitoring and Detection Using XGBoost Algorithm and Image Segmentation

Srinivasan S[1], *Nahadeer Mohamed A*[2], *Somasekar M*[3], *Washim Sulthan M*[4]

[1] *Professor of Practice, Department of Artificial intelligence and Data science, SRM Valliammai Engineering College, Chennai, Tamil Nadu, India.*

[2,3,4]*UG Scholar, Department of Artificial intelligence and Data science, SRM Valliammai Engineering College, Chennai, Tamil Nadu, India.*

Email ID: *srinivasans.ai-ds@srmvalliammai.ac.in*[1], *nahadeermohamed224@gmail.com*[2], *somasekar5011@gmail.com*[3], *wasimsulthan092@gmail.com*[4]

Abstract

Highly sensitive to wildfires are ecosystems, human communities, and climatic balance as well as their respective environments. Inefficiencies, delays, and inadequate coverage define conventional approaches of detection include human observation and satellite imaging. This work proposes a novel artificial intelligence based aerial photography, image processing, and machine learning based wildfire detection system. The system uses edge computing to lower latency and the XGBoost algorithm for exact smoke classification in various environmental circumstances, therefore enabling real-time detection. High-resolution images from stationary cameras, satellites, and drones are analyzed using advanced feature extraction techniques in order to improve detection accuracy. Edge computing eliminates cloud infrastructure and speeds responses by supporting local processing. Early testing shows low false positive rates and good detection accuracy, therefore raising system performance confidence. The scalable suggested approach supports several image sources, therefore supporting extensive wildfire monitoring. Reducing response times and real-time warnings in the suggested solution dramatically improves wildfire management initiatives. It's fit with artificial intelligence and modern computer technology fills in the void between traditional detection techniques and real-time automated solutions.

Keywords: *Artificial Intelligence; Edge computing; Machine Learning; Wildfire Detection*

1. Introduction

Global climate stability, human habitations, and ecological balance are all highly endangered by wildfires. Sophisticated detection and monitoring systems are now required as, in the last few decades, human activity and climate change have led to an explosion of wildfire activity. Ground-level observation and manual reporting are two traditional methods of wildfire detection that are typically not equipped with real-time inputs, and that leads to a response delay and catastrophic loss. Artificial intelligence and image processing have made automated systems viable options for the precise detection and prevention of wildfires. Aerial photography and satellite imaging, among other remote sensing instruments, have transformed environmental monitoring by allowing the monitoring of vast areas continuously. By improving the accuracy of classification and reducing false positives, the integration of machine learning ensemble learning algorithms such as XGBoost has made it possible to improve wildfire detection considerably. Edge computing is also very useful for the real-time processing of data since it lowers latency and allows rapid decision-making. Using these recent technologies will make wildfire detection systems issue timely warnings, thus allowing rapid action and efficient use of resources. [1]

2. Methodology

2.1. Data Formatting

The Toolbox can process spreadsheet or table data. The rows of the table represent data samples. Data variables are table columns. Varying variables may be object attributes or time-stamped measurements. Importantly, every sample has the same variables. Values may be missing, but most should be. Most data is presented in tables. Available data may typically be altered to meet these standards. SOM uses numeric data from the Toolbox, but it can handle both. Categories can be added to data sample labels in the Toolbox. Samples have post it notes attached. The training algorithm ignores them, but the user can refer to them later to determine sample meaning. Handle categorical variables using SOM autolabel. Categories can be turned into numerical variables using mapping or 1-of-n coding for SOM training. Variables must have meaningful number representations to be considered "numeric.": This variable should indicate that B lies between A and C and that the distance between B and A is less than the distance between B and C. Handle identification numbers, error codes, etc., as categorical data because they rarely have meaning. [2-3]

2.2. Construction of Data Sets

Using a MATLAB struct, as described in the Toolbox, you can arrange data imported into MATLAB. Numerical data (.data), text labels (.labels), and metadata on the dataset and its variables are among the several kinds of information this structure compiles. The Toolbox offers several structures, including a map structure that keeps specifics about a Self-Organizing Map (SOM). This structure can be obtained from a numerical matrix by `sD = som_data_struct(D)`. Most functions accept numerical matrices generically; data structures become essential when categorical variables are required. Under such circumstances, these variables are turned into strings and kept in the. labels field as a cell array. [5]

2.3. Data Preprocessing

The Toolbox provides an implementation for only the first of these options. You can equalize histograms and scale numerical variables linearly or logarithmically with Som_normalize. Since the SOM technique calculates vector distances using the Euclidean metric, proper variable scaling is essential. The structure of the map will be largely controlled by a single variable with values ranging from 0 to 1000, as it considerably effects the measured distances. All of the criteria should ideally be equally important. Typically, all variables are linearly scaled to have a single variance. One advantage of using data structures instead of standard data matrices is that they store normalization details within the field. comp_norm, allowing multiple datasets to undergo the same normalization process. Normalization can be repeated for every variable. Picture this: sD contains three numerical variables. All three variables could be linearly scaled to unit variance, the first could be histogram equalized, and the third could be logarithmically scaled. [4]

2.4. Initialization and Training

The correct training approach for an XGBoost model is to train with the appropriate elements. For instance, considering factors like learning rate, number of estimators, and the depth of the decision tree, the model involves many decision trees that work together to improve the estimated value and thus reduce the errors. For rapid learning, date, and immediate implementation, a training model comprising various phases is introduced. In order to achieve a precise training and a better estimate, it is essential to slow down the learning rate. Other factors like tree depth, early halting, and subsampling can be adjusted to have a better influence over the process. These given adjustments help the model function better by reducing the risk of overfitting.

2.5. Visualization and Analysis

There are several methods for visualizing self-organizing maps (SOM). The key tool in the tool set is SOM SHOW, which can display both SOM component planes and U-Matrix. By giving information about the lengths between nearby data points, U-Matrice clears up the cluster formation. A low U-Matrix value indicates the formation of a cluster, while a high value indicates the reach of the cluster's edges. Every part's plane shows the value of one variable for each mapped unit. You could also add labels, data histograms, and paths as more visual tools. Limited tools for quantitative SOM analysis are currently available. SOM QUALITY contains two key metrics: topographic error and average quantization error. In addition, the low-level functions, such as SOM neighbor.

e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 845-848

Table 1 Compares The Performance Metrics of Various Machine Learning Models, Including Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), and Multi-Layer Perceptron (MLP)

Model	Prediction	Precision	Recall	F1	Accuracy
LR	0	0.8130	0.8441	0.8007	0.8041
	1	0.8004	0.8828	0.8308	
RF	0	0.8511	0.8427	0.8161	0.8105
	1	0.8308	0.7963	0.8042	
XGB	0	0.9901	0.9923	0.9954	0.9902
	1	0.9931	0.9910	0.9963	
MLP	0	0.8086	0.8259	0.8171	0.8146
	1	0.8210	0.8033	0.8121	

Figure 1 is the system architecture that demonstrates the integration of edge computing with image segmentation for wildfire detection

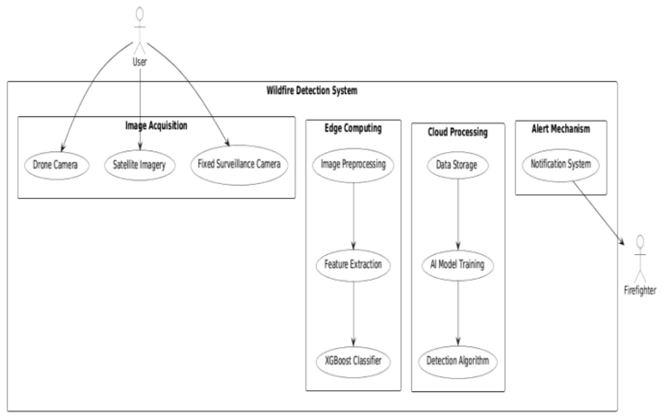


Figure 1 Architecture of the system

3. Results and Discussion

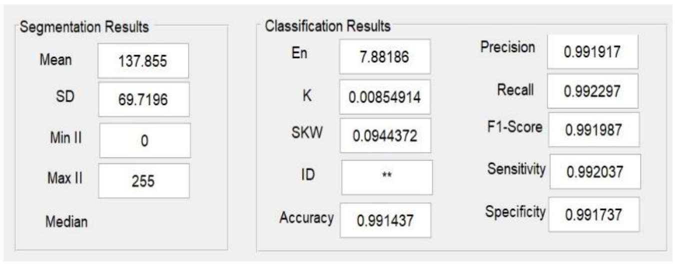
3.1. Results

After experimenting with numerous machine learning models, we achieved an optimal result of 99% using the XGBoost method. This outcome is solely contingent upon the data utilized and may vary for others based on their respective datasets. The

XGBoost algorithm classifies wildfire photos more accurately than other models, which yield a higher rate of false positives. [6]

3.2. Discussion

This conclusion is solely applicable to data pertaining to images. It is not possible to use this technology for live video surveillance or video-based fire detection at this time. In order to detect wildfires, this procedure necessitates the utilization of a specialized storage device for the purpose of storing the photos. (Figure 2)



Segmentation Results

Mean	137.855
SD	69.7196
Min II	0
Max II	255
Median	

Classification Results

En	7.88186	Precision	0.991917
K	0.00854914	Recall	0.992297
SKW	0.0944372	F1-Score	0.991987
ID	**	Sensitivity	0.992037
Accuracy	0.991437	Specificity	0.991737

Figure 2 The Final Output Visualization Showcases the Classified Wildfire Images with Accuracy Metrics Displayed

Conclusion

With only a few false positives, the wildfire monitoring and detection system that is based on image segmentation and the XGBoost algorithm is able to discover the wildfire image with more accuracy. Low-latency detection of the wildfire is made possible by edge computing and machine learning. This research demonstrates how artificial intelligence can be useful in reducing the number of natural disasters, preventing the loss of life, and minimizing the damage that comes from them. Because of this, it is more preventative than the conventional approaches.

e ISSN: 2584-2854
Volume: 03
Issue:03 March 2025
Page No: 845-848

References

[1]. C. C. Chan, S. A. Alvi, X. Zhou, S. Durrani, N. Wilson and M. Yebra, "A Survey on IoT Ground Sensing Systems for Early Wildfire Detection: Technologies, Challenges, and Opportunities," in IEEE Access, vol. 12, pp. 172785-172819, 2024, doi: 10.1109/ ACCESS.2024.3501336.

[2]. C. A. S. Lelis et al., "Drone-Based AI System for Wildfire Monitoring and Risk Prediction," in IEEE Access, vol. 12, pp. 139865-139882, 2024, doi: 10.1109/ACCESS.2024.3462436.

[3]. L. Wang, O. Doukhi and D. J. Lee, "FCDNet: A Lightweight Network for Real-Time Wildfire Core Detection in Drone Thermal Imaging," in IEEE Access, vol. 13, pp. 14516-14530,2025,doi: 10.1109/ ACCESS. 2025. 3526974.

[4]. A.Umunnakwe and K. Davis, "An Optimization of UAV-Based Remote Monitoring for Improving Wildfire Response in Power Systems," in IEEE Open Access Journal of Power and Energy, vol. 10, pp. 678-688,2023,doi: 10.1109/ OAJPE. 2023. 3337760.

[5]. A. Umunnakwe and K. Davis, "A Data-Driven Automated Mitigation Approach for Resilient Wildfire Response in Power Systems," in IEEE Open Access Journal of Power and Energy, vol. 10, pp. 665-677, 2023, doi: 10.1109/OAJPE.2023.3337751.

[6]. E. Casas, L. Ramos, E. Bendek and F. Rivas-Echeverría, "Assessing the Effectiveness of YOLO Architectures for Smoke and Wildfire Detection," in IEEE Access, vol. 11, pp. 96554-96583,2023, doi: 10.1109/ ACCESS. 2023.3312217.