Volume 10, Issue 2, February – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

Real-Time Fire Hazard Monitoring with Deep Learning

Sai Nivedha N.[1]; Dhamotharan R.[2]

M. Sc Data Science[1]; Assistant Professor[2]
[1,2]Department of Computer Science and Information Technology
Kalasalingam Academy of Research and Education Krishnankoil, Tamil Nadu

Publication Date: 2025/03/15

Abstract: Fire outbreaks pose a significant threat to lives and property, making early detection crucial for minimizing damage. Traditional fire detection methods often rely on manual monitoring or conventional image analysis techniques, which can lead to delayed detection and lower accuracy. To address these challenges, this project implements an AI-powered fire detection system using the yolo8 object detection model. The model has been trained on a dataset of 2,509 images, with 1,004 used for training, 754 for validation, and 751 for testing. The system processes video input in real time, detecting fire and marking affected areas with a bounding box and confidence score. Detection details, including the timestamp, fire status, and confidence level, are logged in a CSV file for record-keeping. Additionally, an automated alert system is integrated using Twilio's SMS service, which immediately notifies designated authorities upon fire detection. The model achieves a mean Average Precision (mAP) of 91.3%, a precision of 90.3%, and a recall of 86.9%, demonstrating high reliability in identifying fire incidents. With its ability to detect fire efficiently and provide real-time alerts, this system offers a scalable and effective solution for fire monitoring and prevention.

Keywords: *Computer Vision, Fire Detection, Image Processing, Twilio SMS Notification, Bounding Boxes Detection, Deep Learning.*

How to Cite: Sai Nivedha N.; Dhamotharan R. (2025). Real-Time Fire Hazard Monitoring with Deep Learning. *International Journal of Innovative Science and Research Technology*, 10(2), 2060-2069. https://doi.org/10.38124/ijisrt/25feb1629.

I. INTRODUCTION

Fire detection plays a vital role in ensuring safety and minimizing damage caused by fire hazards. Traditional fire alarm systems primarily rely on smoke, heat, or radiation sensors, which require fire particles to reach the sensor before detection, leading to delays. Additionally, these systems do not provide detailed information such as the fire's location, size, or intensity. With the widespread use of surveillance cameras, integrating vision-based fire detection offers a more effective solution.

Unlike sensor-based systems, image-based detection identifies fire visually in real time without waiting for smoke or heat to spread, significantly reducing response time. A single camera placed at a strategic vantage point can cover large areas, enhancing efficiency compared to conventional sensors, which are more suited for confined spaces.

Additionally, vision-based systems allow authorities to verify incidents through surveillance footage, reducing false alarms. This work focuses on developing a real-time fire detection system using the YOLO object detection model, leveraging deep learning and computer vision techniques.

By processing video input, the system detects fire, highlights affected areas with bounding boxes, logs detection details, and sends automated alerts to authorities, ensuring a faster and more accurate response to potential fire incidents.

II. THE PROPOSED SYSTEM

A. Object Detection Using Yolov8

You Only Look Once (YOLO) is a real-time object detection system that processes an entire image in a single pass through a neural network. The image is divided into multiple regions, and the model predicts bounding boxes along with confidence scores for each detected object.

Here, YOLOv8 is used due to its improved accuracy and speed compared to earlier versions. The model is trained on a custom fire detection dataset using Roboflow, where 2,509 images were labelled and pre-processed. Since commonly used datasets like Common Objects in Context COCO do not include fire detection classes, a custom dataset was necessary. Roboflow 3.0 Object Detection (Fast) was used as the model type, with COCO as a checkpoint to enhance learning.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

The trained model, best.pt, detects fire in real-time by drawing bounding boxes around affected areas with confidence scores. The final model achieves a mean Average Precision (mAP) of 91.3%, a precision of 90.3%, and a recall of 86.9%, ensuring efficient and accurate fire detection suitable for real-time monitoring applications.

III. YOLO v8 ALGORITHM

The process begins with preprocessing the input data, where each video frame undergoes auto orientation correction to ensure proper alignment. Additionally, the frames are resized to 640x640 pixels, maintaining a consistent input size for the model, which helps in improving detection accuracy and computational efficiency.

Once the frames are pre-processed, the YOLOv8 model (best.pt) is loaded. Each video frame is passed through the model, which applies convolutional layers to extract features and uses bounding box regression to locate potential fire regions. The model then classifies the detected objects and assigns a confidence score. If the fire class is identified with a confidence score above 0.5, it is considered a valid detection.

After detecting fire, the system logs detection details such as timestamp, fire status, and confidence score into a CSV file for documentation. To enhance interpretability, a bounding box is drawn around the detected fire region in the video frame.

Additionally, an automated alert system is integrated using Twilio's SMS service. When fire is detected for the first time, the system sends an SMS alert to designated authorities, ensuring a quick response. To prevent redundant alerts, the system ensures that an SMS is sent only once per fire detection event.

The system operates in real-time, continuously analysing video frames until the video ends or a specified time limit is reached. Finally, the system releases the video stream and closes all windows, completing the fire detection process. The inclusion of pre-processing techniques such as auto-orientation correction and resizing further enhances detection reliability and model performance.

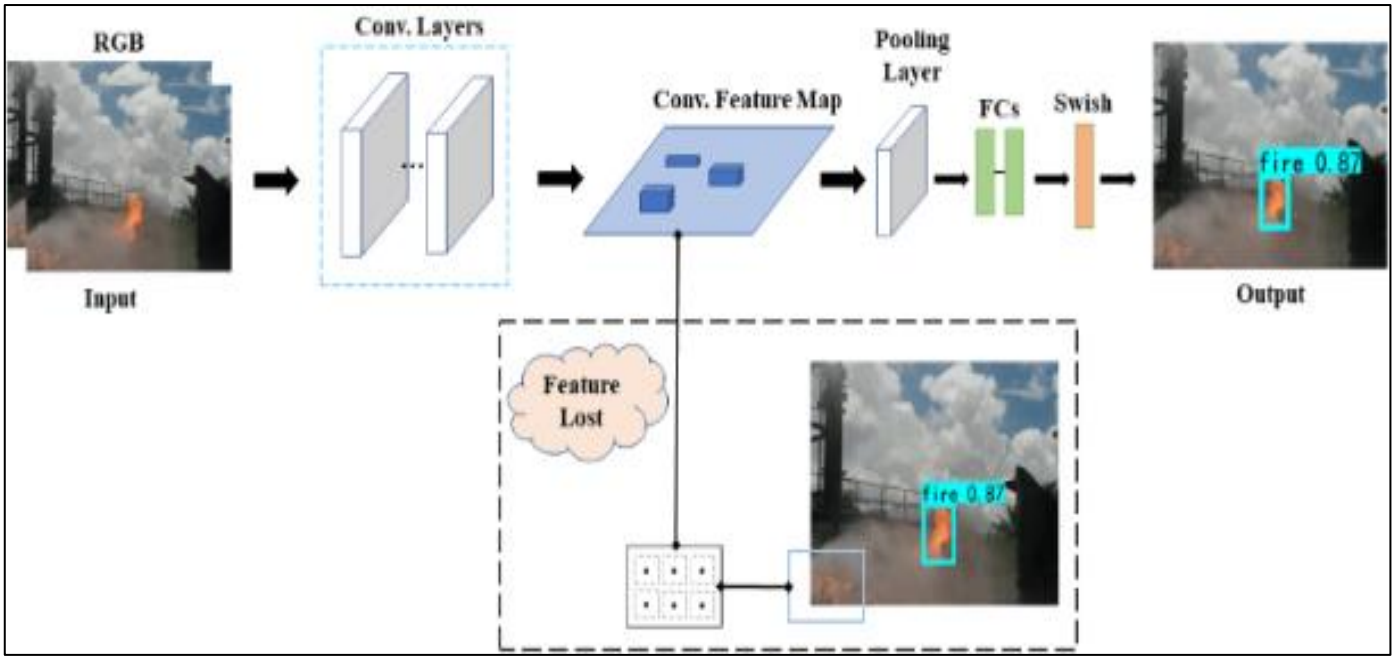


Fig 1: YOLOv8 Algorithm

IV. TRAINING ALGORITHM / METHODOLOGY

This section outlines the methodology for detecting and localizing fire zones using the YOLOv8 model. The system is designed to automatically identify fire patterns in an image and accurately determine their location. The process begins with collecting data from various sources, followed by preprocessing steps such as resizing and annotation. Once the data is prepared, the training phase begins, where a label map is created, and the YOLOv8 model is configured with appropriate parameters. The model is then trained, and key metrics like loss and accuracy are monitored. After training, the testing phase involves adjusting parameters and feeding test images into the model for prediction. The system analyses the images, detects fire zones, and generates bounding boxes around the identified areas. This structured approach ensures reliable detection, making it suitable for real-time fire monitoring applications.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

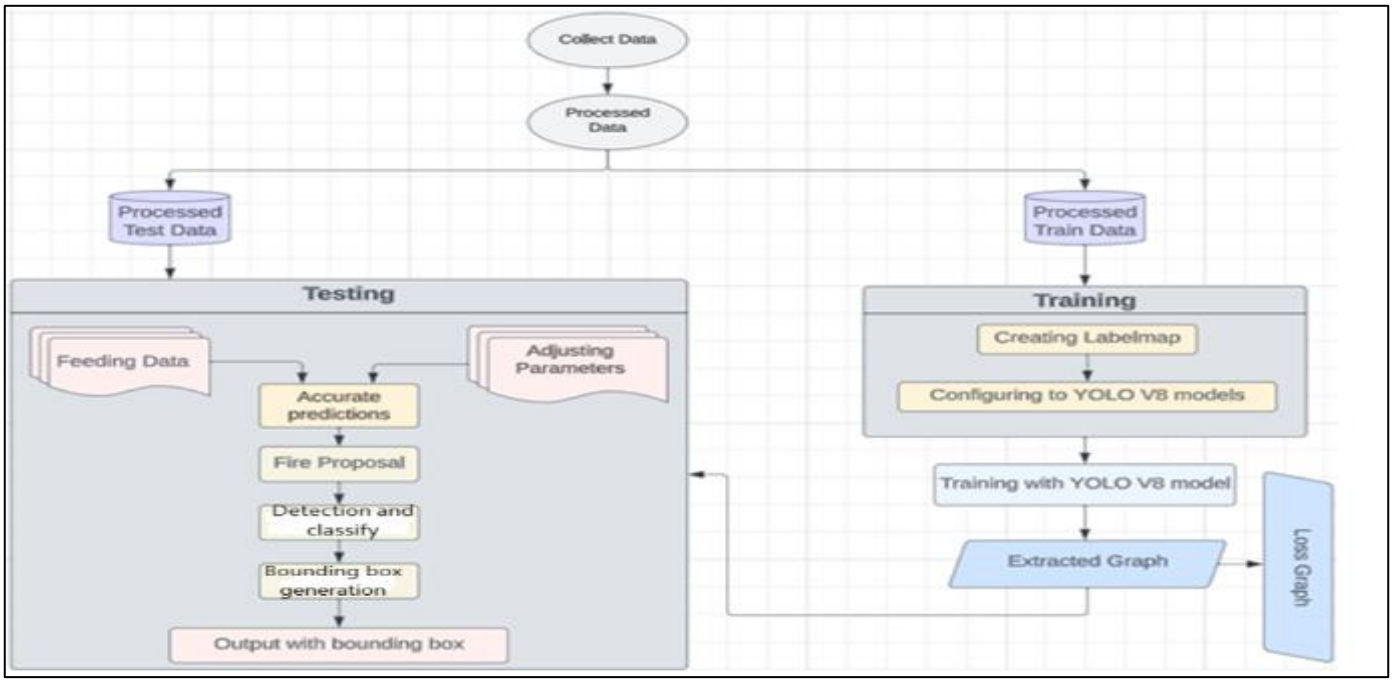


Fig 2: Workflow Architecture Diagram for the Whole Detection Process

A. Image Acquisition

Image acquisition is a crucial step in developing an effective fire detection system. For this study, a total of 2,509 images were collected from various online sources, including crime and emergency response websites, which provide real-world fire incident images. The dataset includes diverse scenarios such as day and night fires, aerial views, fixed-shot fires, surface fires, trunk fires, and canopy fires. The collected images ensure a comprehensive representation of different fire conditions, enhancing the robustness of the detection model.

Table 1: Dataset Statistics

Category	Train	Test	Validation	Total
Fire	602	451	452	1505
Non-Fire	402	300	302	1004
Total	1004	751	754	2509

To maintain a balanced and structured dataset, the images were divided into 1,004 for training, 754 for validation, and 751 for testing. The YOLOv8 model was trained using this dataset, and the best-performing model, best.pt, was selected for further evaluation. This structured dataset and well-defined training process ensure high accuracy in detecting fire in real-time applications.

Fig 3: Samples of Raw Image Data

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

B. Data Pre-Processing

➢ *Augmentation*

Enhancing dataset diversity plays a key role in improving the performance and adaptability of machine learning models. When working with images from various sources, differences in size, resolution, and lighting conditions can impact the training process. To address these variations, different preprocessing techniques were explored and compared to assess their impact on model performance.

One approach involved crop and rotational adjustments, particularly for images where labelling required alignment. Applying rotations within a range of -20 to +20 degrees helped account for real-world variations in object orientation, ensuring that the model could generalize better to different perspectives.

Another aspect considered was brightness correction. Some images had lower visibility due to poor lighting conditions, so adjustments ranging from +20% to -20% in brightness were tested. This step helped improve image clarity, making features more distinguishable for training.

Additionally, flipping techniques were evaluated, where images were randomly flipped to introduce positional variations. This comparison allowed for an understanding of how different preprocessing methods influenced detection accuracy, ensuring a well-prepared dataset for fire recognition.

➢ *Image Resizing*

Standardizing image dimensions is essential for maintaining consistency across the dataset and ensuring optimal model performance. In this process, all images were resized to 640x640 pixels using a stretch-to-fit approach. This resizing method ensures that every image conforms to the required input dimensions while preserving important visual details.

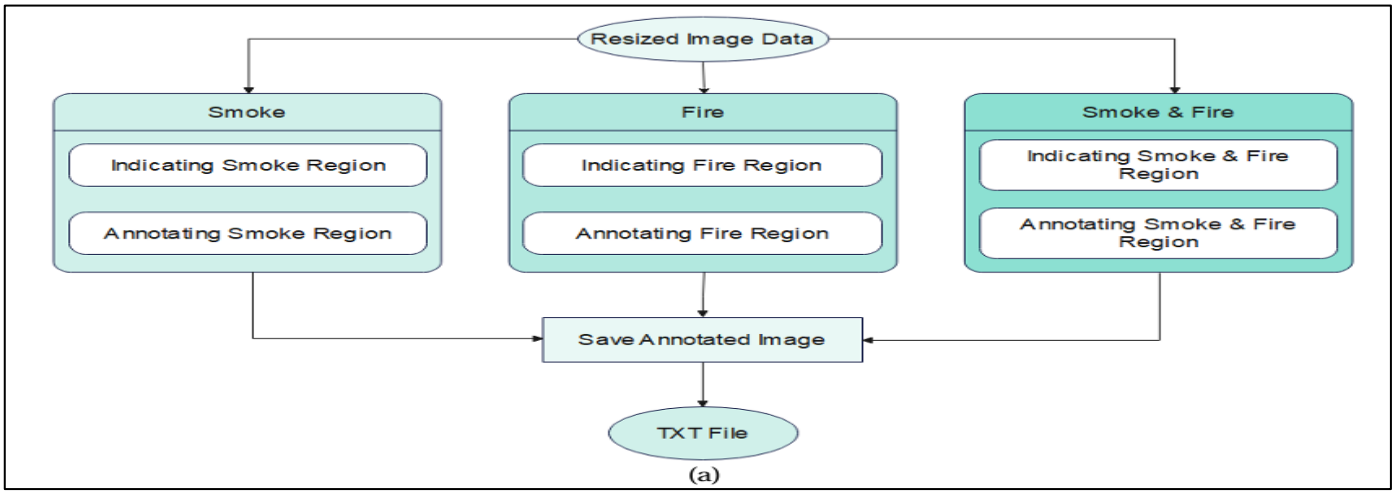
Additionally, auto-orientation was applied to correct any discrepancies in image alignment due to variations in camera angles or metadata inconsistencies. This step helps maintain uniformity across the dataset, preventing potential distortions that could affect the model's ability to accurately detect fire in different scenarios.

Fig 4: Sample of Resized Image Data

➢ *Image Labelling or Annotating*

Roboflow labelling software was used to select the appropriately scaled images. The fire regions in the dataset were annotated using the bounding box tool and polygon tool, ensuring precise labelling. The annotations were automatically saved, and the export option generated a TXT file containing detailed information about the marked fire regions.

Once the labelling process was completed, the dataset underwent a structured data preprocessing workflow. The images were divided into three sets: 40% for training, 30% for testing, and 30% for validation, ensuring a balanced distribution for model development. After pre-processing, the dataset was prepared for further data processing steps to enhance model performance and accuracy.



(a)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

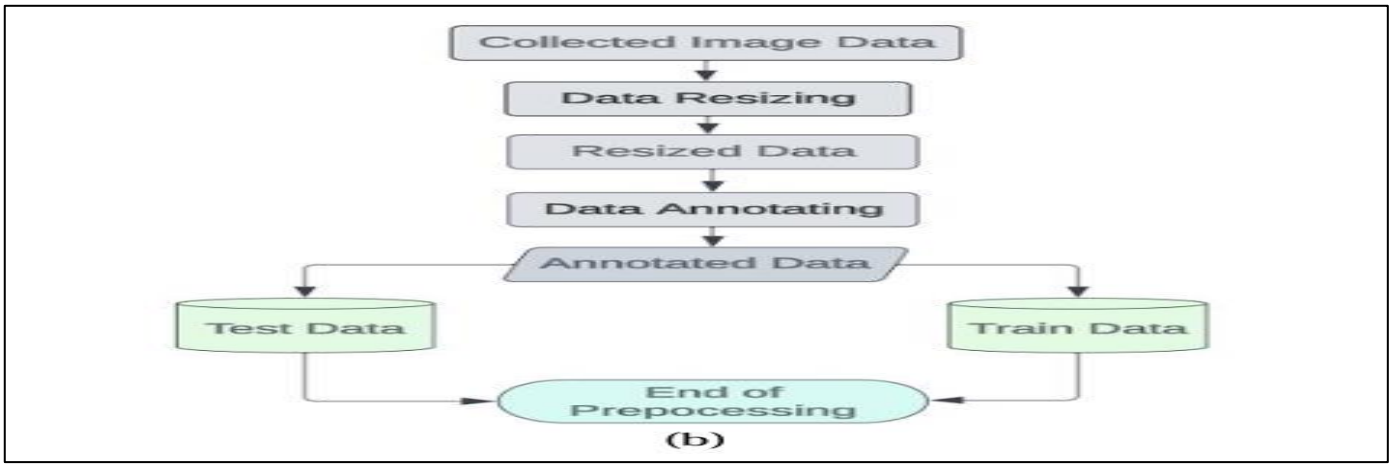


Fig 5: Flowchart for (a) Labelling the Resized Image Data and (b) Data Pre-Processing Steps

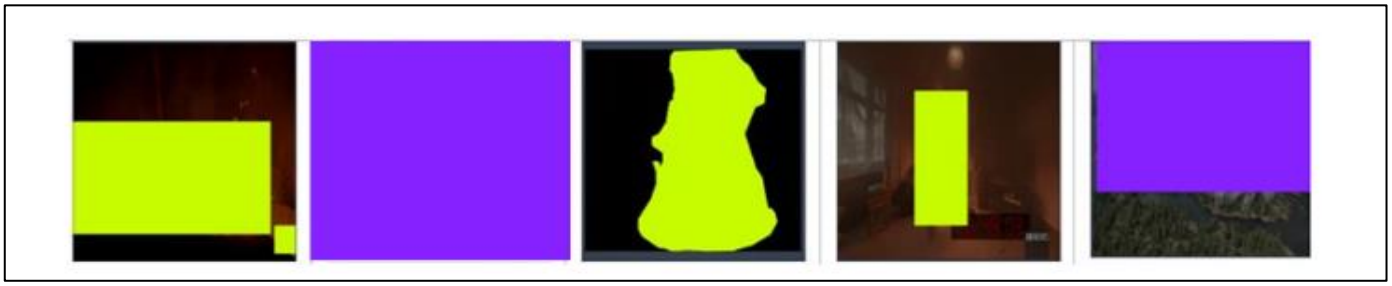


Fig 6: Samples of Labelled Image Data

C. Data Processing

In this data processing context, only one step is considered for generating a TXT file. Considering that, a file of plain text was created for exporting data easily and importing in a structured manner. Then the processed data is set for model training steps.

V. RESULTS AND DISCUSSION

Training the YOLOv8-based fire detection model involved key hyperparameter tuning and optimization. The model was trained for 300 epochs using Stochastic Gradient Descent (SGD) with a batch size of 16, learning rate of 0.01, and weight decay of 0.001.

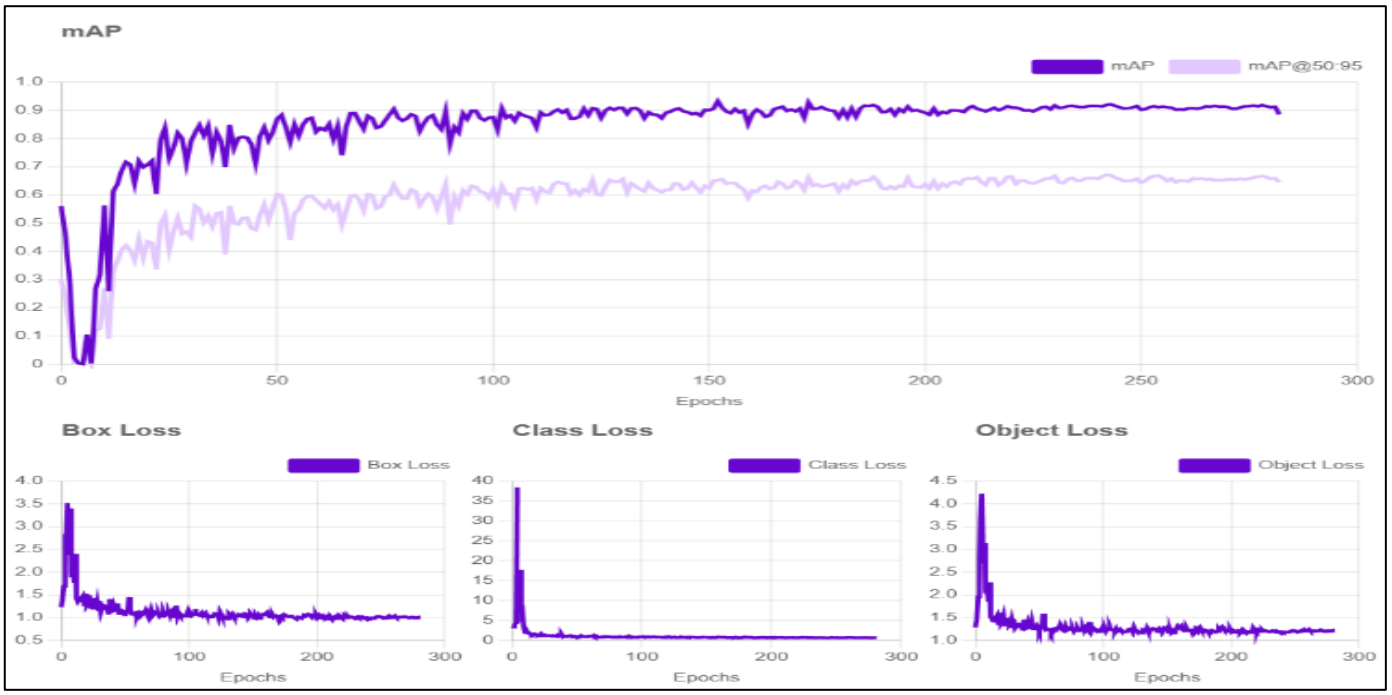


Fig 7: Graphical Representation of Model During Epochs

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

To prevent overfitting, an early stopping mechanism was applied, halting training if no improvement was seen for 50 consecutive epochs. The pre-trained YOLOv8 model helped in faster convergence.

Performance evaluation using mAP@50 showed a steady increase, indicating better fire detection accuracy. The box loss, class loss, and object loss graphs showed a significant decline, confirming effective learning.

A. Model Evaluation

The evaluation metrics employed in this paper to assess the model's performance included precision (P), recall (R), average precision (AP), mean average precision (mAP), F1 score, parameters, floating point operations (FLOPs), and frames per second (FPS). AP represents the area under the precision-recall (PR) curve, while mAP signifies the average AP across different categories. The formulas used for these metrics are outlined as (1)-(3).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} \int_0^1 p(R)\, dR \quad (3)$$

True positive (TP) signifies accurate classification of a sample as positive, while false positive (FP) denotes incorrect classification of a sample as positive. False negative (FN) signifies the misclassification of a sample as negative. 'n' represents the count of categories. FLOPs are a measure of computational complexity, indicating the number of computations performed by a model. FPS stands for frames per second, representing the rate at which frames are transmitted.

The YOLOv8 model was evaluated for fire detection using Google Colab GPU. The model consists of 129 layers, 11,135,987 parameters, and 11,135,971 gradients, with a GFLOP value of 28.6, ensuring efficient computation.

Key performance metrics are as follows:

Table 2: Performance Metrics

Precision	90.3%
Recall	86.9%
mAP@50	91.3%
mAP@50-95	0.661
Fitness Score	0.80

B. Analysis of Results

The results indicate a steady improvement in precision, recall, and mean Average Precision (mAP) over training epochs. The precision and recall curves show an upward trend, suggesting that the model's ability to correctly detect fire and smoke has improved. The loss plots for training and validation exhibit a consistent decline, which indicates effective learning and reduced error over time. The final mAP@50 score is high, reflecting strong object detection performance. However, the mAP@50-95 value is lower, implying that performance varies across different thresholds.

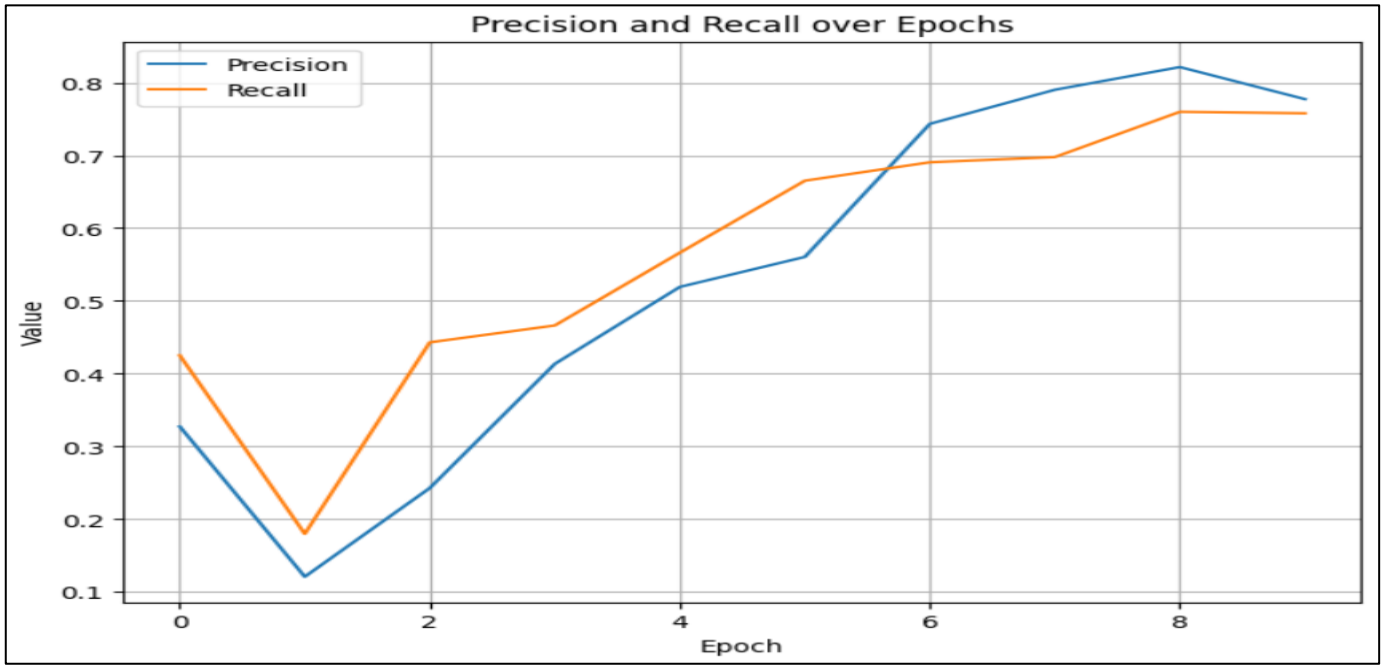


Fig 8: YOLO v8 based Epoch Value

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

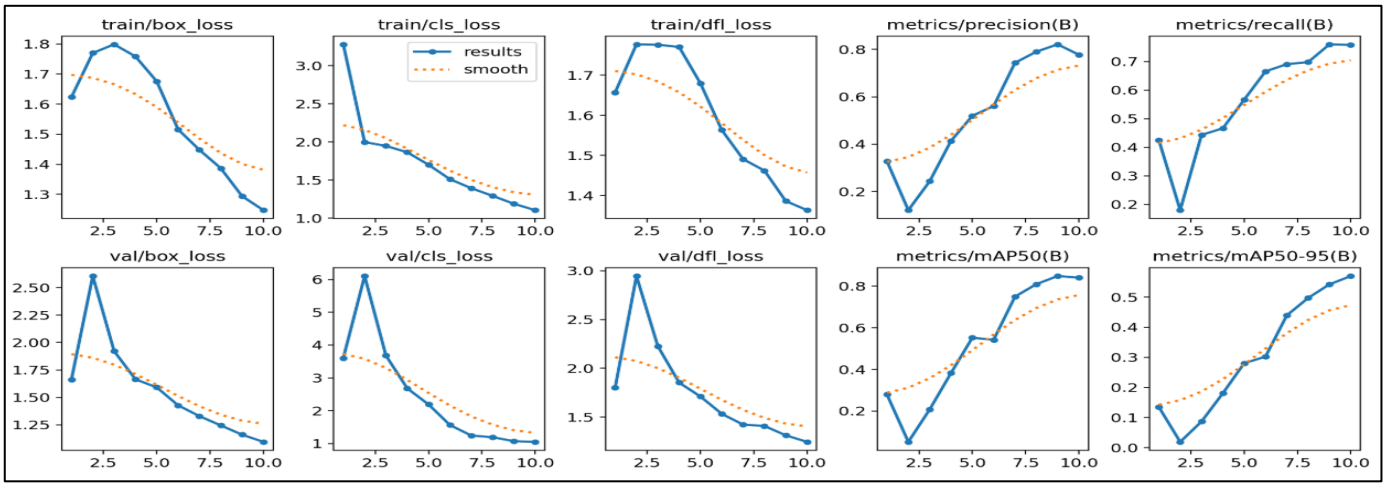


Fig 9: YOLO v8 based Training Graph with 10 Epochs

➢ *Correlation Matrix*

A confusion matrix is a crucial tool in machine learning used to assess the effectiveness of a classification model. It provides insights into the model's strengths and weaknesses by evaluating precision, recall, and overall accuracy. In this case, the matrix defines two key categories: fire and background. True Positives (TP) represent correctly identified instances, whereas False Positives (FP) indicate misclassifications.

The model accurately detects fire in 70% of cases, while 30% of instances are incorrectly classified as background. These values, ranging between 0.01 and 0.70, offer deeper insights into confidence levels for each classification. The findings highlight areas that require improvement, particularly in minimizing false negatives to enhance detection accuracy.

Examining the training performance, the optimal results are observed between the 82nd and 85th iterations. The evaluation of fire detection using the YOLOv8 model is compared against other object detection models, including YOLOv7, YOLOv5, MobileNet-v2, and ResNet-32. The model's performance is analysed across different training iterations, with metrics such as mean Average Precision (mAP@0.5) being monitored to gauge learning effectiveness. A higher mAP@0.5 value indicates superior learning and model efficiency.

Additionally, the F1-score, computed using a structured formula, demonstrates that YOLOv8 consistently surpasses other models. It achieves an F1-score of 60% and a mAP@0.5 value of 57.3%. Further analysis of model complexity highlights YOLOv8's advantage over YOLOv7, which has the highest number of trainable parameters, potentially reducing its ability to generalize effectively.

Moreover, an evaluation of classification accuracy at the 85th iteration indicates an overall performance of 90.45%. The analysis reveals a strong detection capability for fire, achieving 70% accuracy. Despite some classification challenges, YOLOv8 proves to be a promising solution for real-time fire detection across various scenarios.

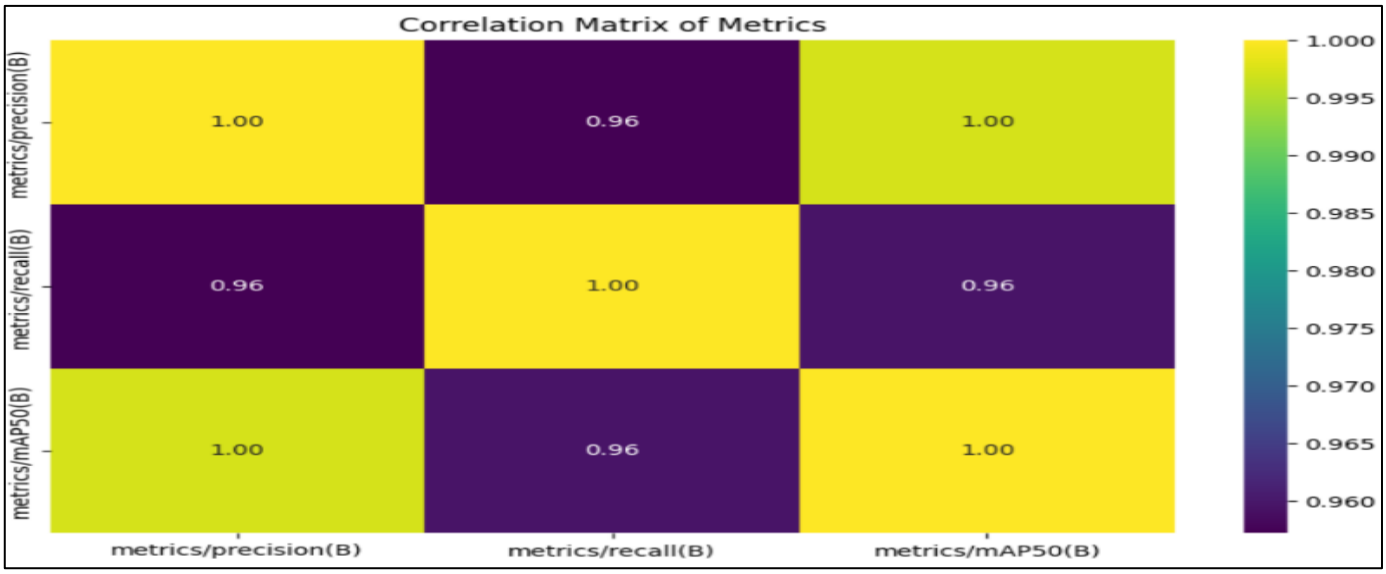


Fig 10: Correlation Matrix of Metrics

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

Table 3: Testing Execution of YOLOv7, YOLOv5, MobileNetv2 and ResNet-32

Model	Epoch	Class	Trainable Parameters	F1 Score	mAP@0.5
Proposed YOLO v8	50	All	11.13M	0.587	0.555
Proposed YOLO v8	85	All	11.13M	0.607	0.573
YOLO v7	50	All	37.2M	0.430	0.372
YOLO v7	85	All	37.2M	0.458	0.391
YOLO v5	50	All	7.2M	0.487	0.426
YOLO v5	85	All	7.2M	0.487	0.426
MobileNet- v2	50	All	3.4M	0.359	0.324
MobileNet- v2	85	All	3.4M	0.365	0.335
ResNet- 32	50	All	0.47M	0.267	0.245
ResNet- 32	85	All	0.47M	0.276	0.255

C. Visualization

The model underwent training for 100 epochs, with each epoch representing a full pass through the training dataset. Throughout this process, the model's parameters were continuously updated based on computed losses and gradients. Training was concluded at 85 epochs, as optimal results were observed around the 80th step, in line with the predefined configuration settings. Early stopping was not activated since the criteria for halting training were not met within the specified 85 epochs. The entire training process took approximately three hours, though this duration could vary depending on computational resources and hardware capabilities, highlighting the resource-intensive nature of deep learning training.

Performance peaked at 85 epochs before showing signs of decline, a typical case of overfitting where the model becomes too specialized in the training data and loses its ability to generalize effectively to new inputs. A comparative analysis of various models revealed that YOLOv8 consistently outperformed others in fire detection tasks. Specifically, YOLOv8 achieved detection accuracies of 90% and 51%, whereas models such as YOLOv7, ResNet-32, and MobileNet-v2 failed to detect fire in certain cases.

Further assessment demonstrated that YOLOv8 consistently provided superior detection accuracy compared to YOLOv7, YOLOv5, ResNet-32, and MobileNet-v2. For instance, when evaluating multiple images, the detection rates for YOLOv8 reached 79%, whereas YOLOv7, YOLOv5, ResNet-32, and MobileNet-v2 recorded lower accuracies of 60%, 47%, 27%, and 49%, respectively. Overall, YOLOv8 proved to be the most effective model for fire detection, outperforming other architectures trained on the same dataset. While training across different epochs showed minor improvements, the YOLOv8 model trained for 85 epochs delivered the best performance across all evaluation metrics. Beyond this point, extending the training to 100 or 150 epochs led to a decline in accuracy due to overfitting.

Fig 11: Sample Detected Images

D. Real Time Fire Detection

The real-time fire detection system processes incoming images and detects fire with a confidence score using deep learning techniques. It identifies fire regions by drawing bounding boxes around them and assigns a confidence value. Upon detection, the system sends an alert message to a designated person, ensuring quick response. Additionally, a log entry is created, recording the timestamp of the fire event along with the confidence score, providing a record for future analysis and verification.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

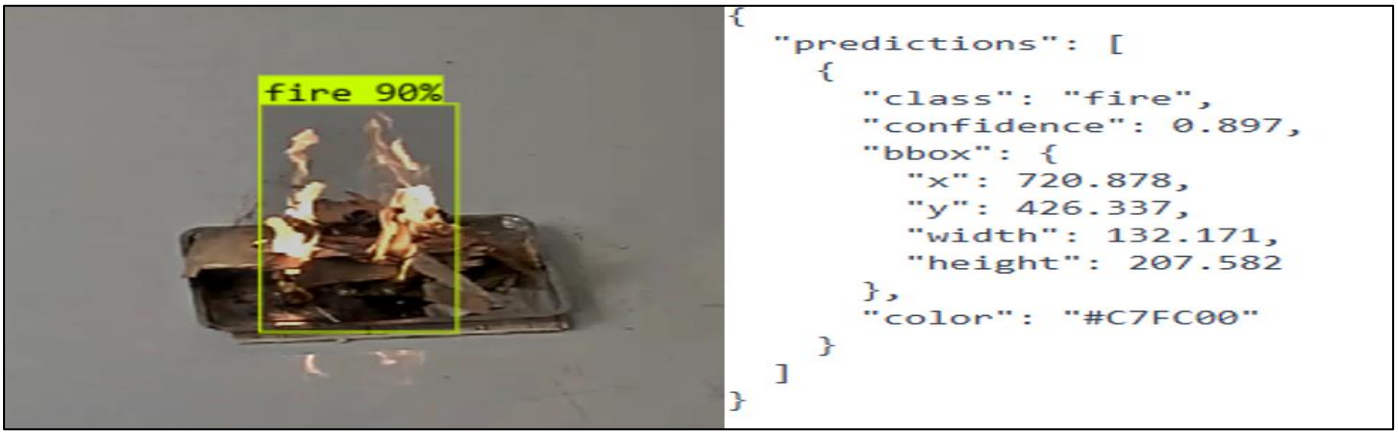


Fig 12: Real Time Fire Detection with the Predictions

Timestamp	Fire Detected	Confidence
28-02-2025 12:34	Yes	0.50366229
28-02-2025 12:34	Yes	0.76005715
28-02-2025 12:34	Yes	0.57605851
28-02-2025 12:34	Yes	0.56210381
28-02-2025 12:34	Yes	0.86243713

Fig 13: Sample of Saved Log after Completing the Process

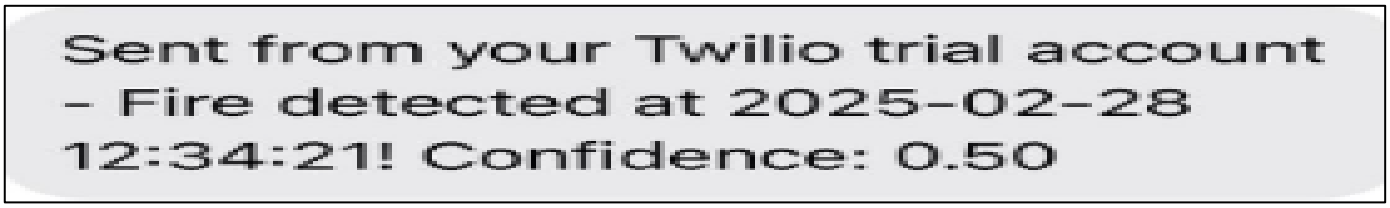


Fig 14: Text Message of Fire Detection via SMS

Fig 15: Process Completion Acknowledgement Text

VI. CONCLUSION

Fire outbreaks present a major risk to lives and property, necessitating an efficient and reliable detection system. This project successfully developed an AI-powered fire detection system using the YOLOv8 object detection model, which significantly outperforms traditional methods in terms of speed and accuracy. The model was trained on a dataset of 2,509 images, split into training, validation, and testing sets, ensuring robust learning and generalization. Unlike conventional image analysis techniques, which may result in delayed detection, this system processes video input in real time, marking fire-affected areas with bounding boxes and confidence scores. Additionally, a CSV-based logging system records critical details such as timestamps, fire status, and confidence levels, while an automated SMS alert system using Twilio ensures immediate notification to designated authorities upon fire detection.

A comparative analysis was conducted with YOLOv7, YOLOv5, ResNet-32, and MobileNet v2 to evaluate the effectiveness of the proposed model. While other models struggled with lower detection accuracy and inconsistencies, YOLOv8 demonstrated superior performance with a mean Average Precision (mAP) of 91.3%, a precision of 90.3%, and a recall of 86.9%. The model's training peaked at 85 epochs, where it achieved optimal detection capability without overfitting. These results highlight YOLOv8's efficiency in real-time fire detection across various scenarios, making it a practical solution for fire monitoring and prevention. Future improvements could focus on optimizing the model for edge devices and enhancing its

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25feb1629

robustness under varying environmental conditions to ensure wider applicability and deployment.

REFERENCES

[1]. S. Rahman, J. H. Rony, J. Uddin, and M. A. Samad, "Real-time obstacle detection with YOLOv8 in a WSN using UAV aerial photography," J. Imaging, vol. 9, no. 10, p. 216, Oct. 2023, doi: 10.3390/jimaging9100216.

[2]. R. Siddiqua, S. Rahman, and J. Uddin, "A deep learning-based dengue mosquito detection method using faster R-CNN and image processing techniques," Ann. Emerg. Technol. Comput., vol. 5, no. 3, pp. 11–23, Jul. 2021, doi: 10.33166/AETiC.2021.03.002.

[3]. S. B. Hasan, S. Rahman, M. Khaliluzzaman, and S. Ahmed, "Smoke detection from different environmental conditions using faster R-CNN approach based on deep neural network," in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, LNICST, vol. 325 LNICST, 2020, pp. 705–717. doi: 10.1007/978-3-030 52856-0_56.

[4]. P. D. K. He, G. Gkioxari and R. Girshick, "Mask R-CNN," in IEEE international conference on computer vision, IEEE, 2017, pp. 2961–2969.

[5]. Khandaker, and J. Uddin, "Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique," Int. Arab J. Inf. Technol., vol. 17, no. 6, pp. 947–953, Nov. 2020, doi: 10.34028/iajit/17/6/13.

[6]. R. A. Khan, J. Uddin, and S. Corraya, "Real-time fire detection using enhanced color segmentation and novel foreground extraction," in 4th International Conference on Advances in Electrical Engineering, ICAEE 2017, IEEE, Sep. 2017, pp. 488–493. doi: 10.1109/ICAEE.2017.8255405.

[7]. R. A. Khan, J. Uddin, S. Corraya, and J.-M. Kim, "Machine vision-based indoor fire detection using static and dynamic features," Int. J. Control Autom., vol. 11, no. 6, 2018, doi: 10.14257/ijca.2018.11.6.09.

[8]. H. Zheng, J. Duan, Y. Dong, and Y. Liu, "Real-time fire detection algorithms running on small embedded devices based on MobileNetv3 and YOLOv4," Fire Ecol., vol. 19, no. 1, p. 31, May 2023, doi: 10.1186/s42408-023-00189-0.

[9]. H. Du, W. Zhu, K. Peng, and W. Li, "Improved high speed flame detection method based on YOLOv7," Open J. Appl. Sci., vol. 12, no. 12, pp. 2004–2018, 2022, doi: 10.4236/ojapps.2022.1212140.

[10]. S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y. I. Cho, "An improved wildfire smoke detection based on YOLOv8 and UAV images," Sensors (Basel)., vol. 23, no. 20, 2023, doi: 10.3390/s23208374.