Proceedings of the 5th International Conference on Communication, Computing and Electronics Systems (ICCCES-2026)
DVD Part Number: CFP25VH5-DVD; ISBN: 979-8-3315-5620-4

Enhancing Car Accident Detection: A Fuzzy Logic-based Evaluation of Faster R-CNN and YOLOv8

1st Shaik Basheer
Dept. of Computer Science and Engineering
SRM University AP
Neerukonda, Andhra Pradesh-522240
Email: shaikbasheer_c@srmap.edu.in

2nd Chanchal Biswas
Dept. of Computer Science and Engineering
SRM University AP
Neerukonda, Andhra Pradesh-522240
Email: chanchal_biswas@srmap.edu.in

3rd Dunga Priyatham
Dept. of Electronics and Communication and Engineering
SRM University AP
Neerukonda, Andhra Pradesh-522240
Email: priyatham_dunga@srmap.edu.in

4th Shaik Abdul Karim
Dept. of Computer Science and Engineering
SRM University AP
Neerukonda, Andhra Pradesh-522240
Email: karim_shaik@srmap.edu.in

5th Abdul Rehan
Dept. of Computer Science and Engineering
SRM University AP
Neerukonda, Andhra Pradesh-522240
Email: rehan_abdul@srmap.edu.in

***Abstract*—Real-time accident detection is essential for enhancing road safety and enabling quick emergency response. Models such as Faster R-CNN and YOLOv8 already offer strong performance, achieving detection accuracies of 80.1% and 99.5% (mAP@0.5), respectively. However, in challenging real-world scenarios—such as low visibility, occlusions, and overlapping objects—these models may produce inconsistent or conflicting predictions.To overcome this limitation, we propose a hybrid accident detection system that combines YOLOv8 and Faster R-CNN using a fuzzy logic-based decision mechanism. Confidence scores from both models are processed through fuzzy membership functions—Ignore, Uncertain, and Strong Detection—and a rule-based inference engine to produce a more stable and interpretable final output.Experiments on a custom dataset containing Car, Car Accident, and Fire Accident classes show that the hybrid approach achieves an improved mAP@0.5 of 90.3%, reduces false positives, and delivers more reliable classification. These results demonstrate that fuzzy logic fusion effectively enhances the consistency and robustness of accident detection, making the system suitable for intelligent surveillance and real-time traffic monitoring.**

Index Terms—YOLOv8, Faster R-CNN, Accident Detection, Fuzzy Logic, Intelligent Transportation Systems, Real-Time Surveillance.

I. INTRODUCTION

A. Brief Introduction

Intelligent Transportation Systems (ITS) require fast and accurate accident detection, whereas manual monitoring is often slow and unreliable [2]. Deep learning models such as YOLOv8 provide high-speed and reliable detection even under difficult conditions such as poor lighting or heavy traffic [3], [4]. Enhanced YOLOv8 variants that use BiFPN, GAM, improved loss functions, and small-object detection heads further boost performance [5], [6], [7]. Lightweight versions like TP-YOLOv8 also support edge deployment [9].

This work builds on these advancements by integrating YOLOv8 and Faster R-CNN through fuzzy logic to achieve more consistent and accurate accident detection in real-time scenarios.

B. Project Objectives

- Develop a real-time detection system for accidents and fires.
- Compare YOLOv8 and Faster R-CNN in terms of accuracy, speed, and reliability.
- Use fuzzy logic to resolve uncertain or conflicting predictions.
- Reduce false positives using a hybrid decision-making approach.
- Validate the system on real accident footage under different conditions.
- Support quicker and more accurate emergency response mechanisms.

C. Project Significance

This work enhances accident detection by integrating deep learning with fuzzy logic to reduce false alarms and improve decision reliability. Traditional methods often misinterpret events, delaying emergency response. The proposed hybrid fusion of YOLOv8 and Faster R-CNN produces more consistent predictions, strengthening surveillance systems, improving detection speed, and contributing to safer road environments.

Section I introduces the problem and motivation. Section II describes the proposed architecture. Section III reviews related work. Section IV explains dataset preparation, model training, and fuzzy integration. Section V presents results and performance analysis. Section VI visualizes the fuzzy decision surface and final classification.

II. PROPOSED WORK

This study proposes a hybrid accident detection model combining YOLOv8 and Faster R-CNN with fuzzy logic. Fuzzy logic merges the confidence scores from both detectors to produce a stable final decision.

The system uses three membership categories: Ignore (low confidence), Uncertain (moderate confidence), and Strong Detection (high confidence). This ensures smooth transitions between decisions and improves reliability in challenging conditions like occlusions, overlapping objects, or low visibility.

Overall, the framework reduces false detections, handles cases where one detector may fail, and provides consistent, robust real-time accident detection.

III. LITERATURE REVIEW

Daxin Tian et al. [3] propose an accident detection system using Cooperative Vehicle-Infrastructure Systems (CVIS), achieving a detection time of 0.0461 seconds with 90.02% AP.

Akshaya et al. [6] use YOLOv8 and ultrasonic sensors for proactive collision avoidance, demonstrating reliable obstacle detection for highway safety.

Girija M et al. [8] use deep learning to predict traffic accidents based on historical data, showing AI's potential in risk assessment.

Xingyu Liu et al. [5] introduce BGS-YOLO, which uses background subtraction to improve object detection in dynamic traffic scenes.

Jiahui Chen et al. [10] develop an improved YOLOv8 model for IIoT-based accident detection, achieving higher efficiency and accuracy.

Vasileios Efthymiou et al. [1] compare CNN architectures for vehicle detection in V2X environments, analyzing detection accuracy and speed.

Nitheesh Vijayan et al. [4] study collision detection using machine learning with a focus on sensor fusion and predictive modeling.

Jun Peng et al. [7] improve YOLOv8 with better feature extraction for autonomous driving applications.

Bharathi Mohan G et al. [2] present an AI-based real-time accident detection and emergency response system using IoT integration.

Zhaole Ning et al. [9] introduce TP-YOLOv8, a lightweight model with attention mechanisms for efficient traffic accident recognition.

Richa Singh et al. [17] develop a Faster R-CNN-based accident detection system suitable for surveillance.

Trung-Nghia Le et al. [15] propose Attention R-CNN for improved accident detection using attention mechanisms.

Mr. Shankar K et al. [13] explore YOLOv8 improvements for safer terrain monitoring.

Sheli Sinha Chaudhuri et al. [16] compare YOLO and Faster R-CNN for vehicle detection in autonomous driving.

R. Vasanthi et al. [14] demonstrate YOLOv8 for real-time car damage detection.

Prathilothamai M et al. [11] design a YOLOv8-based system for detecting traffic violations.

Siddhant Mishra et al. [12] focus on vehicle damage detection using deep learning and discuss related challenges.

IV. METHODOLOGY

A. Research Design

The study focuses on real-time accident detection using machine learning models with fuzzy logic. YOLOv8 and Faster R-CNN classify three classes: Car, Car Accident, and Fire Accident. The Fuzzy Inference System (FIS) combines confidence scores from both models to produce a reliable final decision.

B. Data Collection Preprocessing

1) **Dataset:** Images of Car, Car Accident, and Fire Accident collected from real-time traffic cameras.
2) **Annotation:** Labeled using Roboflow with bounding boxes. Dataset split:
   - Training: 80
   - Validation: 4
   - Test: 16
3) **Preprocessing:** Correct orientation and resize to 640 x 640 pixels for consistent input.
4) **Augmentation:** Flipping, rotation, brightness/contrast adjustment, and Gaussian noise to improve generalization.

These steps ensure a robust dataset suitable for real-world accident detection.

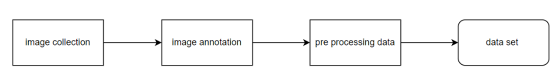

Fig. 1: Flow of data Pre-processing

As shown in Fig. 1, the preprocessing pipeline standardizes the dataset through orientation correction, resizing, and augmentation to ensure consistent model input.

C. Model Training

1) YOLOv8: YOLOv8 is an enhanced and faster iteration of the YOLO series, offering improved accuracy and efficiency. It uses a streamlined single-stage architecture that integrates feature extraction, object localization, and classification. With an upgraded backbone, optimized feature fusion, and multi-scale prediction, YOLOv8 effectively detects objects of varying sizes. Additionally, it provides multiple model variants to balance speed and accuracy depending on hardware capabilities, making it a flexible and robust choice for real-time object detection tasks.

D. YOLOv8 Framework

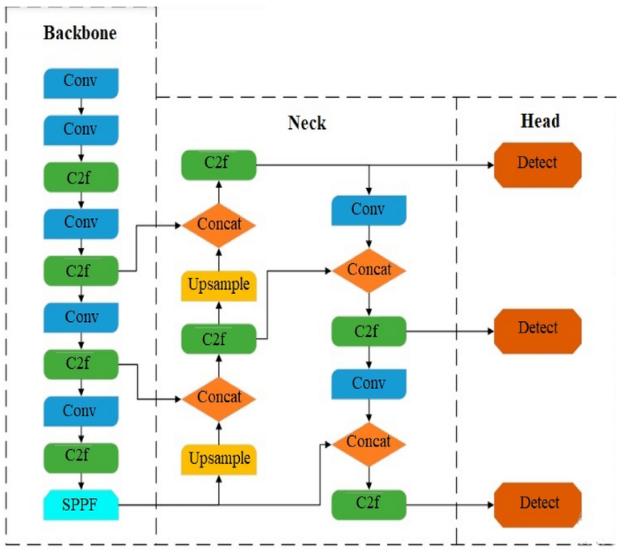

Fig. 2: YOLOv8 framework.

As shown in Fig. 2, the framework includes feature extraction, detection heads, and prediction layers for bounding boxes and class probabilities.

AAs mentioned above, YOLOv8 consists of three primary components in its architecture:

- Backbone
- Neck
- Head

The Backbone serves as the main convolutional network responsible for extracting key features from input images. YOLOv8 employs a customized CSPDarknet53 with cross-stage partial connections to enhance information flow and improve detection accuracy.

The Neck aggregates feature maps from multiple backbone layers, enabling effective multi-scale object detection.

Instead of a standard FPN, YOLOv8 uses the C2f (Cross-Stage Fusion) module, which combines detailed spatial features with high-level semantic information, boosting detection performance, particularly for small objects.

The Head generates the final outputs, including bounding box coordinates, objectness scores, and class probabilities, which are then integrated to provide accurate real-time object detection results.

E. YOLOv8 Training

The YOLOv8 model was trained on the dataset with the following configurations:

- Learning rates: 0.01, 0.003, 0.0001
- Batch sizes: 8, 16, and 32
- Optimizers: Adam, AdamW, SGD, and RMSprop
- Momentum: 0.8 and 0.95
- Weight decay: 0.0001 and 0.01

YOLOv8 employs three primary loss functions:

- Bounding box regression: CIoU Loss
- Classification: Binary Cross-Entropy (BCE) Loss
- Objectness: Binary Cross-Entropy (BCE) Loss

Evaluation metrics include mAP (mean Average Precision), Precision, Recall, and F1-score.

Training was performed on Google Colab with GPU acceleration, and the trained models were stored in Google Drive for further use. The following libraries were utilized for the YOLOv8 model:

- Ultralytics: Main library for training, evaluating, and deploying YOLOv8 models
- OpenCV-Python: For image processing
- NumPy: For numerical computations
- Matplotlib: For visualizations and plotting results

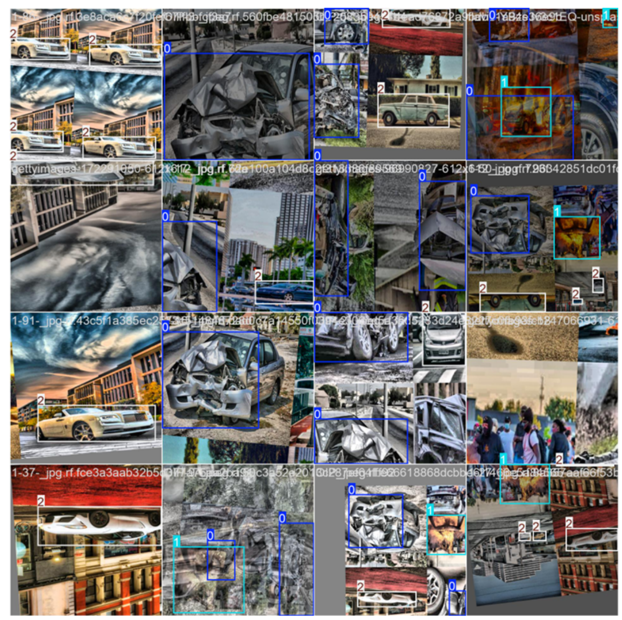

Fig. 3: Training images generated through YOLOv8 model.

As shown in Fig. 3, the training images are generated with augmentations to improve model accuracy.

F. Faster R-CNN

The Faster R-CNN is a two-stage object detector that combines a Region Proposal Network (RPN) with a Fast R-CNN detector. An input image is first passed through a CNN backbone (e.g., ResNet or VGG) to extract high-level features.

The RPN generates candidate regions (anchors) of different sizes and aspect ratios, classifying them as object or background and refining their bounding boxes.

The Region of Interest (ROI) Pooling layer then extracts fixed-size feature maps from these proposals, which are passed to the Fast R-CNN head. The head predicts the object class, confidence score, and refined bounding box for each detected object.

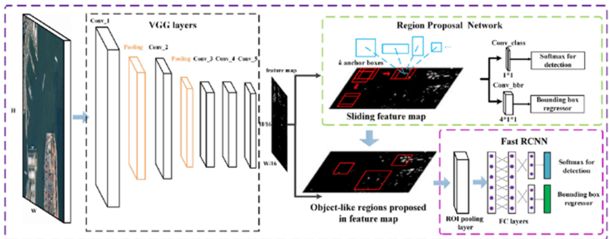

Fig. 4: Faster R-CNN framework.

As shown in Fig. 4, the Faster R-CNN framework consists of a region proposal network, feature extraction layers, and detection heads for bounding box and class prediction.

G. Faster R-CNN Training

The Faster R-CNN backbone is based on ResNet-50 with a Feature Pyramid Network (FPN), which enhances multi-scale feature representation for improved detection performance.

1) Region Proposal Network (RPN) Settings:

- The RPN, an integral part of Faster R-CNN, generates Regions of Interest (ROIs) for object detection.
- It uses anchor boxes to propose candidate regions.
- Default RPN parameters (from torchvision) include:
  - Anchor sizes: [32, 64, 128, 256, 512]
  - Aspect ratios: [0.5, 1.0, 2.0]
  - Pre-NMS proposals: 2000 (training), 1000 (inference)
  - NMS threshold: 0.7
  - Batch size for RPN proposals: 256

2) Training Configuration:

- Batch sizes: 8, 16
- Learning rates: 0.0001, 0.003, 0.01
- Optimizers: Adam, AdamW, SGD, RMSprop
  - Betas: (0.95, 0.999)
  - Weight decay: 0.0001
- Learning rate scheduler: StepLR (Step size: 3, Gamma: 0.1)
- Loss functions:
  - Classification Loss: Cross-Entropy
  - Bounding Box Regression Loss: Smooth L1
  - RPN Objectness Loss
  - RPN Box Regression Loss

Libraries used for Faster R-CNN training include:

- Torch (PyTorch)
- Torchvision
- NumPy
- Matplotlib
- torch.utils.data.DataLoader
- Tqdm

Different fuzzy-rule settings were tested, and the final Low/Medium/High confidence ranges were selected based on the best validation results and stable F1-scores.

H. Fuzzy Decision-Making Methodology

The fuzzy inference system is designed to process two main input types:

- Detected Class Labels: Predictions from YOLOv8 and Faster R-CNN, such as “Car Accident”, “Fire Accident”, or “Car”.
- Confidence Scores: Numerical confidence values between 0 and 1 associated with each prediction.

These inputs are fuzzified to interpret uncertainty and variation in model predictions.

1) Fuzzy Membership Functions and Rule Design: **Input Membership Functions:**

- Low Confidence (0 to 0.5) – Indicates weak detection.
- Medium Confidence (0.3 to 0.7) – Indicates moderately reliable detection.
- High Confidence (0.5 to 1) – Indicates strong and likely correct detection.

Output Membership Functions:

- Strong Detection: Confident classification of the detected event.
- Uncertain: Ambiguous detection due to weak or conflicting predictions.
- Ignore: Detection is unreliable and disregarded.

Fuzzy Rules:

- Rule 1: If YOLOv8 detects “Car Accident” with High confidence AND Faster R-CNN detects “Car Accident” with Medium confidence, THEN Final Decision = Car Accident (Strong Detection).
- Rule 2: If one model predicts with Low confidence, the final decision is Uncertain.
- Rule 3: If both models have Medium confidence, THEN Final Decision = Uncertain.
- Rule 4: If both models detect with High confidence, THEN Final Decision = Strong Detection.
- Rule 5: If both models detect with Low confidence, THEN Final Decision = Ignore.

These rules allow the fuzzy system to balance agreement and uncertainty for reliable classification.

I. Fuzzy System Implementation

- Tool Used: scikit-fuzzy in Python
- Defuzzification Method: Centroid
- Edge Case Handling:
  - If models predict different classes, the output is Uncertain.
  - If one model has High confidence and the other Low confidence, the output is Uncertain.

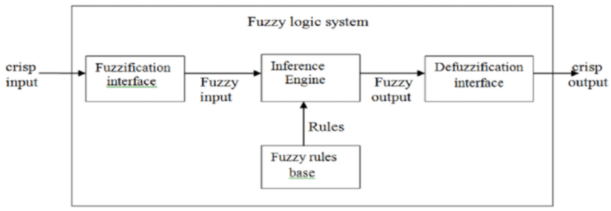

Fig. 5: Fuzzy-Logic architecture.

As shown in Fig. 5, the fuzzy module fuses confidence scores from YOLOv8 and Faster R-CNN.

J. Proposed Model Architecture

The proposed framework is a hybrid system that integrates predictions from YOLOv8 and Faster R-CNN using a fuzzy logic-based decision mechanism. Each detector processes the input image independently, generating class labels along with confidence scores. These outputs are subsequently input into the fuzzy logic module, which handles uncertainty and applies predefined rules to determine the final decision. By combining the strengths of both models, the system achieves more accurate, reliable, and interpretable results for real-time accident detection.

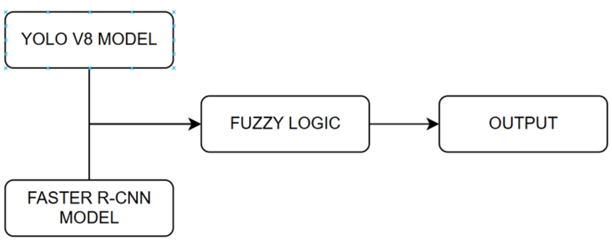

Fig. 6: Proposed model architecture.

Fig. 6 shows the hybrid framework combining YOLOv8 and Faster R-CNN via fuzzy logic for real-time accident detection Experimental results show that YOLOv8 generally gives higher confidence scores than Faster R-CNN. Using fuzzy logic merges their strengths, producing more consistent and reliable decisions. This combines YOLOv8's speed with Faster R-CNN's accurate localization for interpretable detection outputs.

Each frame is analyzed independently, so video tracking metrics (MOTA/MOTP) were not included and will be added in future work.

V. EVALUATION & PERFORMANCE METRICS

To assess model performance, the following metrics were analyzed:

- Object Detection Models (YOLOv8 & Faster R-CNN): mAP (mean Average Precision)

TABLE I: Best combinations performed on the parameters of YOLOv8 model

Optimizer	LR	BS	Mom.	WD	mAP50	mAP50-95
Adam	0.0001	32	0.8	0.0001	0.995	0.531
Adam	0.0001	8	0.95	0.01	0.931	0.463
Adam	0.0001	32	0.8	0.01	0.897	0.459
AdamW	0.0001	32	0.8	0.01	0.956	0.563
AdamW	0.0001	32	0.8	0.0001	0.945	0.501
AdamW	0.0001	16	0.95	0.01	0.934	0.556
SGD	0.0001	32	0.8	0.0001	0.897	0.521
SGD	0.0001	16	0.8	0.01	0.888	0.541
SGD	0.0001	32	0.95	0.01	0.871	0.508
RMSprop	0.0001	32	0.95	0.0001	0.653	0.421
RMSprop	0.0001	16	0.95	0.01	0.568	0.382
RMSprop	0.0001	8	0.8	0.0001	0.556	0.334

Although YOLOv8 achieves a very high **mAP@50 of 99.5%**, the mAP50–95 value is lower because stricter IoU thresholds require more precise bounding-box alignment. In complex accident scenes with occlusions and irregular object shapes, localization accuracy slightly decreases, causing the drop in overall mAP50–95. This reflects bounding-box precision limitations rather than classification issues.

This table presents the mean Average Precision (mAP) results for the YOLOv8 model. From a total of 144 parameter combinations across different optimizers, the top three results from each optimizer were selected for comparison. The model achieving the highest accuracy was trained using the Adam optimizer, reaching 99.5% mAP@50.

Overview on the ADAM optimizer with the parameters below:

- Learning rate 0.001
- Batch size 32
- Momentum 0.8
- Weight 0.0001

Results are MAP50 0.995 and MAP50-99 0.531.

TABLE II: Performance of different optimizers on YOLOv8 model

Optimizer	Learning rate	Batch size	Momentum	Weight	MAP
Adam	0.0001	4	0.8	0.0001	0.801
AdamW	0.0001	4	0.8	0.01	0.785
SGD	0.0001	16	0.8	0.01	0.723
RMSprop	0.0001	16	0.8	0.01	0.566

These represent the top results obtained for each optimizer based on the tested parameters. The highest-performing model was trained using the Adam optimizer, achieving an accuracy of 80.1% (as reported earlier for Faster R-CNN).

A. Precision

Precision quantifies the correctness of positive predictions. It is defined as the ratio of true positive predictions to the total number of predicted positives.

- Formula: Precision = TP / (TP + FP)

B. *Precision*

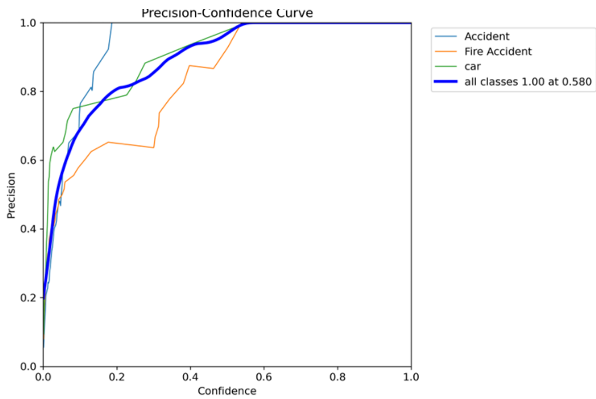

Fig. 7: Precision curve.

As shown in Fig. 7, the precision curve represents the model's ability to correctly identify positive predictions across confidence thresholds.

C. *Recall*

Recall (Sensitivity) evaluates the model's ability to correctly identify positive instances. It is calculated as the ratio of true positives to the total actual positives:

- Formula: Recall = TP / (TP + FN)

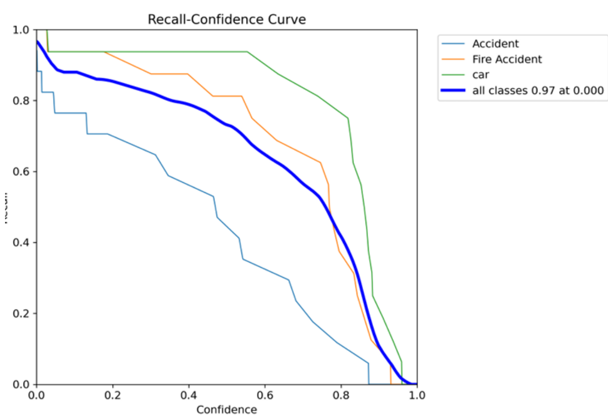

Fig. 8: Recall curve.

The Recall-Confidence Curve for the proposed hybrid model shows how recall varies across different confidence thresholds for the classes: Accident, Fire Accident, and Car. The model achieves a maximum recall of 0.97 at a confidence threshold of 0.0, with Car consistently achieving the highest recall.

D. *Precision vs Recall*

The precision-recall curve illustrates detection performance for the three classes. The model achieves the highest accuracy for Car (AP: 0.973), followed by Fire Accident (AP: 0.932), and Accident (AP: 0.803). The bold blue curve represents the mean Average Precision (mAP@0.5) of 0.903.

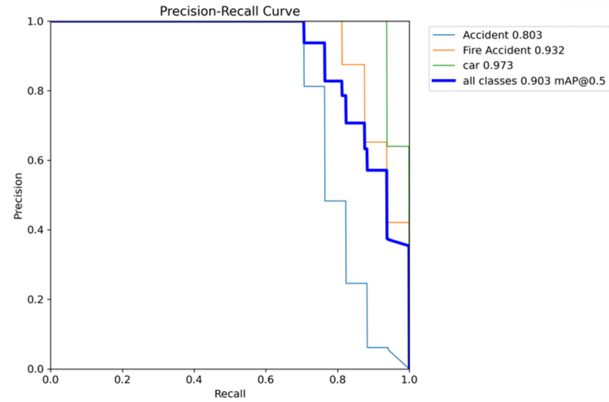

Fig. 9: Precision-Recall Curve.

E. *F1-Score*

The F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure especially useful in cases of class imbalance:

- Formula: F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

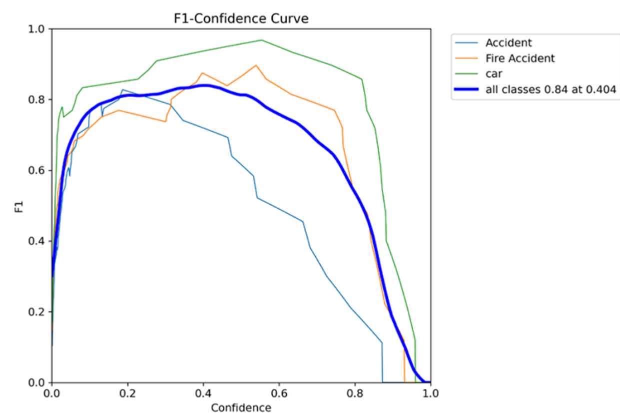

Fig. 10: F1-Score Curve.

As shown in Fig. 10, the F1-Score curve demonstrates balanced detection performance across all classes, highlighting the effectiveness of the proposed hybrid model.

The F1-Confidence Curve indicates that the optimal trade-off between precision and recall occurs at a confidence threshold of 0.404, where the overall F1 score peaks at 0.84. Among individual classes, Car demonstrates the highest and most stable F1 performance, consistently approaching 0.9.

F. *Confusion Matrix*

A confusion matrix summarizes a classification model's performance by comparing predicted and actual values.

G. Confusion Matrix

H. Confusion Matrix

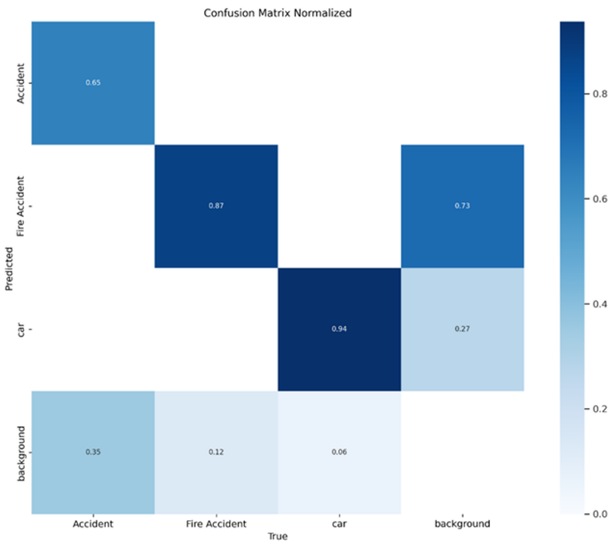

Fig. 11: Confusion Matrix of the hybrid model.

Fig. 11 shows strong detection for Car (94%) and Fire Accident (87%), with lower performance for Accident (65%).

As shown in Fig. 11, the hybrid model achieves strong detection performance for Car (94% accuracy) and Fire Accident (87% accuracy). The Accident class shows lower performance, correctly identifying 65% of instances, with 35% misclassified as Background. Additionally, 27% of Background instances are incorrectly predicted as Car.

I. Loss Functions

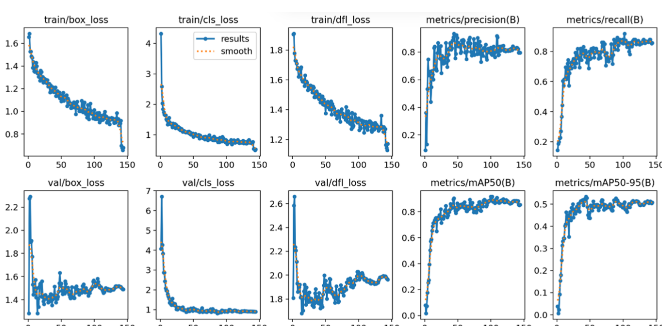

Fig. 12: Training and validation loss curves.

Fig. 12 illustrates the convergence of losses during training, indicating stable optimization.

- train/box loss: Represents the error in bounding box regression, decreasing from above 1.6 to approximately 0.8.
- train/cls loss: Classification loss during training starts above 4 and drops rapidly below 1.
- train/dfl loss: Distribution Focal Loss decreases from around 1.9 to about 1.2.
- metrics/precision(B): Precision rises quickly during early training and stabilizes around 0.9.
- metrics/recall(B): Recall improves gradually and levels off near 0.9.
- val/box loss: Validation box loss reduces and stabilizes around 1.4–1.5.
- val/cls loss: Validation classification loss drops sharply from nearly 7 to around 1.
- val/dfl loss: Validation DFL stabilizes at a slightly higher value, around 1.8–2.0.
- metrics/mAP50(B): Rises rapidly and plateaus near 0.85.
- metrics/mAP50-95(B): Increases steadily to approximately 0.55.

J. Real-Time FPS Performance

To evaluate the practical usability of the proposed hybrid model in surveillance systems, the real-time processing speed was measured. The hybrid system achieves an average of **24–26 FPS** on a 30 FPS video stream, which is sufficient for real-time monitoring. Individually, YOLOv8 runs at approximately **42 FPS**, while Faster R-CNN operates at around **7 FPS**. The fuzzy fusion module introduces less than **1 ms** overhead, ensuring that the overall system maintains real-time performance.
Confidence intervals and cross-validation were not used because of the limited dataset size. Future versions will include these for better statistical validation.

VI. Final Fuzzy Decision & Surface

The hybrid system uses fuzzy logic to combine confidence scores from YOLOv8 and Faster R-CNN for more reliable detection. The fuzzy decision surface shows how confidence levels from both models affect the final output. Three membership functions are used: Ignore (low confidence), Uncertain (moderate confidence), and Strong Detection (high confidence). This approach smoothly handles ambiguity and improves robustness in challenging scenarios.

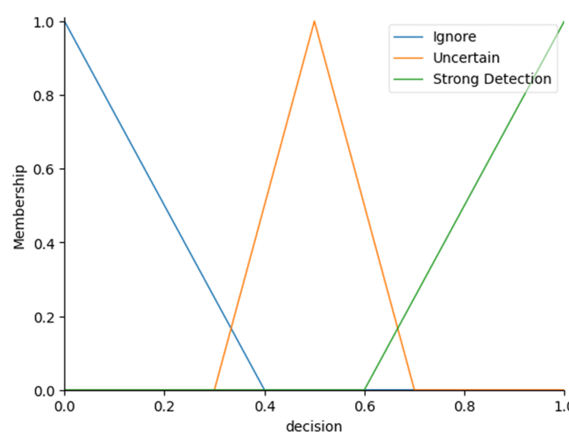

Fig. 13: Final decision graph using fuzzy logic.

Fig. 13 shows the final decision outputs of the hybrid model after fuzzy logic fusion.

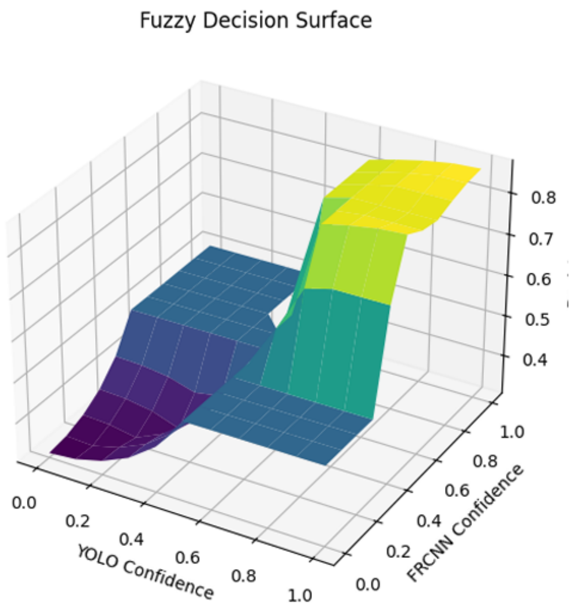

Fig. 14: Fuzzy decision surface visualization.

Fig. 14 visualizes how the fuzzy logic module maps confidence scores from YOLOv8 and Faster R-CNN to final detection decisions.

As shown in Fig. 14, the fuzzy decision surface illustrates how the final decision changes according to confidence values from both models.

VII. Conclusion

The fuzzy logic fusion of YOLOv8 and Faster R-CNN enhances reliability and accuracy in real-time accident detection. By managing uncertainty through fuzzy inference, the hybrid model reduces false positives and demonstrates robust performance under challenging visual conditions. The reported mAP@0.5 of 0.903 confirms the effectiveness of the approach, particularly for Car and Fire Accident classes, while highlighting areas for improvement in general Accident detection.

References

[1] V. Efthymiou, S. Basagiannis and S. Petridou, "CNN Architectures for Vehicle Detection in V2X Environments: A Comparative Study," 2024 IEEE Conference on Standards for Communications and Networking (CSCN), Belgrade, Serbia, 2024, pp. 113-119, doi: 10.1109/CSCN63874.2024.10849688.

[2] B. M. G, P. R. Sanikommu and G. V. Teja, "AI-Enhanced Real-Time Accident Detection with Smart Emergency Response System," 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT), Gwalior, India, 2024, pp. 1-6, doi: 10.1109/INSPECT63485.2024.10896184.

[3] D. Tian, C. Zhang, X. Duan and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," in IEEE Access, vol. 7, pp. 127453-127463, 2019, doi: 10.1109/ACCESS.2019.2939532.

[4] N. Vijayan, S. Pandey and P. R M, "Collision Detection and Prevention for Automobiles using Machine Learning," 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-4, doi: 10.1109/TQCEBT59414.2024.10545080.

[5] Liu, Xingyu, et al. "Enhancing Intelligent Road Target Monitoring: A Novel BGS YOLO Approach Based on the YOLOv8 Algorithm." IEEE Open Journal of Intelligent Transportation Systems (2024).

[6] Kavitha, M., et al. "Proactive Highway Collision Avoidance using YOLOv8 and Ultrasonic Sensors." 2024 13th International Conference on System Modeling Advancement in Research Trends (SMART). IEEE, 2024.

[7] Peng, Jun, et al. "Road Object Detection Algorithm Based on Improved YOLOv8." 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2024.

[8] Girija, M., and V. Divya. "Road traffic accident prediction using deep learning." 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS). IEEE, 2024.

[9] Ning, Zhaole, et al. "TP-YOLOv8: a lightweight and accurate model for traffic accident recognition." The Journal of Supercomputing 81.4 (2025): 1-31.

[10] Chen, Jiahui, et al. "Surveillance-Based Road Accident Detection Method Using Improved YOLOv8 for IIoT Applications." 2024 IEEE/CIC International Conference on Communications in China (ICCC). IEEE, 2024.

[11] Prathilothamai, M., et al. "Traffic Rules Violation Detection System using YOLO v8 Model." 2024 10th International Conference on Electrical Energy Systems (ICEES). IEEE, 2024.

[12] Mishra, Siddhant, and Divya Kamal. "Vehicle Damage Identification using Deep Learning Techniques." 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). IEEE, 2024.

[13] Shankar, K., et al. "YOLOv8-Driven Integration of Advanced Detection Technologies for Enhanced Terrain Safety." 2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI). IEEE, 2024.

[14] Yogeshwaran, S., and R. Vasanthi. "YOLOv8-Powered Real-Time Car Damage Detection." 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). IEEE, 2025.

[15] Le, Trung-Nghia, et al. "Attention R-CNN for accident detection." 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020.

[16] Maity, Madhusri, Sriparna Banerjee, and Sheli Sinha Chaudhuri. "Faster R-CNN and YOLO based vehicle detection: A survey." 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2021.

[17] Sharma, Yashika, and Richa Singh. "Smart vehicle accident detection system using Faster R-CNN." 2021 10th International Conference on System Modeling Advancement in Research Trends (SMART). IEEE, 2021.