IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 11, Issue 05, May 2024 https://www.ijirae.com/archives

Car Accident Detection and Alert System Leveraging Deep Learning

Padma R,
Department of Computer science and Engineering,
Ramaiah University of Applied Sciences,
Bengaluru, INDIA
r.padma2@gmail.com / padma.cs.et@msruas.ac.in
Shilpa H H, Banupriya P
Department of Computer science and Engineering,
Ramaiah University of Applied Sciences,
Bengaluru, INDIA
Shilpahh204@gmail.com / shilpa.cs.et@msruas.ac.in
banupriya.csee@gmail.com
Srikanth V, Yashwanth T M, Adarsha C N
Department of Computer Science and Engineering
Ramaiah University of Applied Sciences
Bengaluru, INDIA
srikanthv25501@gmail.com, yashwanth342001@gmail.com, aaditya0147@gmail.com

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.11/Issue05/MYAE10102
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.11/Issue05/MYAE10102
Received: 20, April 2024 | Revised: 14, May 2024 | Accepted: 23, May 2024 | Published Online: 30, May 2024
https://www.ijirae.com/volumes/Vol11/iss-05/24.MYAE10102.pdf
Article Citation: Padma,Shilpa,Banupriya,Srikanth,Yashwanth,Adarsha(2024). Car Accident Detection and Alert System Leveraging Deep Learning. IJIRAE:: International Journal of Innovative Research in Advanced Engineering, Volume 11, Issue 05 of 2024 pages 598-610
Doi:> https://doi.org/10.26562/ijirae.2024.v1105.24
BibTeX Key Padma@2024Car
Academic Editor-Chief: Dr.A.Arul Lawrence Selvakumar, AM Publications, India

Abstract: This paper presents a novel approach to address the critical issue of car accidents through an advanced Accident Detection and Alerting System (ADAS) empowered by cutting-edge technologies in deep learning, computer vision, and communication APIs. With road safety being a paramount concern worldwide, timely detection of car accidents followed by swift alerting mechanisms can significantly reduce the severity of accidents and save lives. The proposed system architecture incorporates the Ultralytics framework, Convolutional Neural Networks (CNN), You Only Look Once version 8 (YOLOv8), OpenCV, and Twilio API. Initially, YOLOv8 is employed to perform real-time detection of vehicles and potential accident scenarios from live video feeds obtained from roadside cameras or vehicle-mounted cameras. Subsequently, CNN models are utilized for comprehensive analysis of detected objects to discern potential accidents based on predetermined criteria, including sudden deceleration, collision patterns, and anomalous vehicle movements. Upon successful detection of an accident, the system triggers an alert mechanism via the Twilio API, which delivers instantaneous notifications to emergency services, nearby medical facilities, and designated contacts of the involved parties. These alerts are enriched with crucial details such as the accident location, vehicle types involved, and severity assessments derived from CNN analyses. Furthermore, the integration of OpenCV facilitates preprocessing tasks such as noise reduction, image enhancement, and object tracking, thereby enhancing the system's robustness and accuracy in diverse environmental conditions and lighting scenarios. The proposed ADAS offers numerous benefits, including real-time accident detection, automated alerting of emergency services, and minimized response times, ultimately contributing to mitigating the severity of accidents and preserving lives. By harnessing the power of deep learning algorithms, advanced computer vision techniques, and seamless communication APIs, this system represents a significant step towards fostering safer road environments within smart cities and transportation networks.

Keywords: OpenCV, CNN, Yolov8, Ultralytics, Deep learning

I. INTRODUCTION

Road accidents remain a significant global concern, accounting for a substantial number of fatalities and injuries each year. Despite advancements in vehicle safety technology and infrastructure improvements, the timely detection of accidents and rapid response mechanisms are crucial to mitigating their severity and saving lives. Traditional accident detection methods often rely on human observation or post-incident reporting, leading to delays in emergency response and increased risk to accident victims. In response to these challenges, advanced technologies such as deep learning, computer vision, and real-time communication APIs have emerged as promising solutions for enhancing road safety.

This paper introduces a comprehensive Car Accident Detection and Alerting System (CADAS) that harnesses the power of deep learning techniques, specifically the Ultralytics framework, Convolutional Neural Networks (CNN), You Only Look Once version 8 (YOLOv8), OpenCV, and the Twilio API. The integration of these technologies enables the development of an intelligent system capable of autonomously detecting car accidents in real-time and promptly alerting relevant authorities and stakeholders. The foundation of the CADAS lies in YOLOv8, a state-of-the-art CNN-based object detection algorithm renowned for its efficiency and accuracy in real-time object recognition tasks. By leveraging YOLOv8 within the Ultralytics framework, our system achieves robust and reliable detection of vehicles and potential accident scenarios from live video streams obtained from roadside or vehicle-mounted cameras. Furthermore, the utilization of CNN models enables comprehensive analysis of detected objects to identify potential accidents based on predefined criteria, such as sudden changes in vehicle trajectories, collision patterns, and anomalous behavior. This deep learning approach enhances the system's ability to discern genuine accident situations from false alarms, thereby optimizing emergency response efforts. In addition to its detection capabilities, the CADAS incorporates OpenCV for image preprocessing tasks, including noise reduction, image enhancement, and object tracking. These preprocessing steps are crucial for improving the system's performance in varying environmental conditions and ensuring accurate detection under challenging circumstances. Moreover, the integration of the Twilio API enables seamless communication with emergency services, nearby medical facilities, and designated contacts of the involved parties. Real-time alerts generated by the CADAS include vital information such as the accident location, types of vehicles involved, and severity assessments derived from CNN analyses, facilitating swift and targeted response efforts. In summary, the CADAS represents a significant advancement in road safety technology, offering a proactive approach to accident detection and response. By combining deep learning algorithms, advanced computer vision techniques, and real-time communication capabilities, our system aims to reduce the severity of car accidents, minimize response times, and ultimately save lives on our roads.

II. LITERATURE SURVEY

Nicky Kuttukkaran et al. [4] proposed heartbeat a major factor for differentiating accidents then preferred Bluetooth for sending alert signals. This approach is limited by the range of Bluetooth and heartbeat is not a suitable measurement for accident verification because it is subjective to interpretation. Sanjay Kumar Singh [5] studied road accidents in India at state, city and rural levels to find out that 50% of the cities are prone to road accident fatalities and the number of accidents is estimated to rise to 250,000 by 2025. Chaitali Khandbahale et al. [6] performed multi-object detection using embedded detection for accident prevention.

This approach has the limitation of high economic cost and the embedded system comprises sensors that may not function properly during rainy forecasts. Nedjet Dogru et al. [7] applied a clustering algorithm for grouping vehicles based on speed and location. This perspective will only give data about the pattern of movement and the model did not provide optimum accuracy at the time of deployment. Vipul Rana et. al [8] proposed accidental prevention through output and user side experience by asking users to submit a form about the type of climate, type of vehicles, etc. This verification is limited to those cases where the victim can fill the form. Deeksha Gour et al. [9] proposed optimized YOLO working on a system for accident detection. Possibility of misclassification of the required target object that can lower the efficiency of the system. Xi Jianfeng et al.[10] adopted support vector machine (SVM) for accident detection which is found to be limited while localizing objects and instance segmentation during detection hence instance classification is hindered. Raad Ahmed Hadi specifically studied the use of background subtraction, feature-based tracking and region-based tracking methods in the application of vehicle detection [11]. Zhong-Qiu Zhao et al. [12] proposed generic object detection to classify existing objects in an image using masking and recurrent neural networks. This method reduces preprocessing time as well as enhances the pipeline for detection and validation and it can further be customized for vehicle detection. Kaiming He et al. [13] extended fast R-CNN using masking to perform instance segmentation that can be potentially used for custom object detection applications including car accidents. Sodikov et al. [14] studied the accidents occurring in Uzbekistan to provide a visualization of the traffic accident distribution with respect to the day, week, month and year from which we can draw insights about the accident frequency and severity of the road accidents in any countryin general. S. Ramya et al. [15] studied various parameters affecting road accident severity and found out that Random Forest Classifier outperforms its counterparts in handling accident severity prediction due to its ability to handle seasonality in accidental vehicles and casualties. Jabar H. Yousif et al. [16] presented an approach to construct an optimal neural network for traffic accident severity prediction with a proper feature selection to handle any correlations between the severity of accidents and weather conditions in Jordan and can be extended to other countries as well.

Various machine learning techniques have been widely used to detect car crashes based on different kinds of video data. Ki (2007) proposes a vision-based traffic accident detection system using charge coupled device (CCD) cameras in order to detect, record, and report traffic accidents automatically. Vishnu and Rajalakshmi (2016) exploit linear discriminant analysis (LDA) and SVM for monitoring traffic using live video files from surveillance cameras. Ravindran et al. (2016) propose a novel supervised learning model based on machine vision techniques and five SVMs trained with histogram of oriented gradients (HOG) and gray-level co-occurrence matrix (GLCM) features, which successfully detect road accidents from static images. Arceda and Riveros (2018) propose a car crash detection system that combines a violent flow (ViF) descriptor and SVMs using closed-circuit television (CCTV) video data, which are detected through CNN. State-of-the-art deep learning techniques have been recently applied to car crash detection using video data. Chan et al. (2017) offer a dynamic-spatial-attention (DSA) recurrent neural network (RNN) model to predict car accidents from dashboard cameras.

Their model is trained to distribute soft-attention to object candidates gathering subtle cues dynamically as well as modeling the temporal dependencies of all cues to robustly predict accidents. Naidenov & Sysoev (2019) develop a car accident detecting system based on CNN using video capture recordings. Yao et al. (2019) present an unsupervised deep learning framework for traffic accident detection using dashboard cameras. In particular, their approach can detect traffic accidents by predicting traffic participant trajectories as well as their future locations. CrashCatcher (Wagner-Kaiser, 2020) is one well-known car crash detection model that uses video data from dashboard cameras. A hierarchical recurrent neural network (HRNN), which is based on two different layers of long short-term memory (LSTM), is used to train the model. In particular, the first layer analyzes a time series of video data from dashboard cameras. The second layer encodes the results of the first layer and trains the model using the labeled video data (i.e., crash data and non-crash data).

III. METHODOLOGY

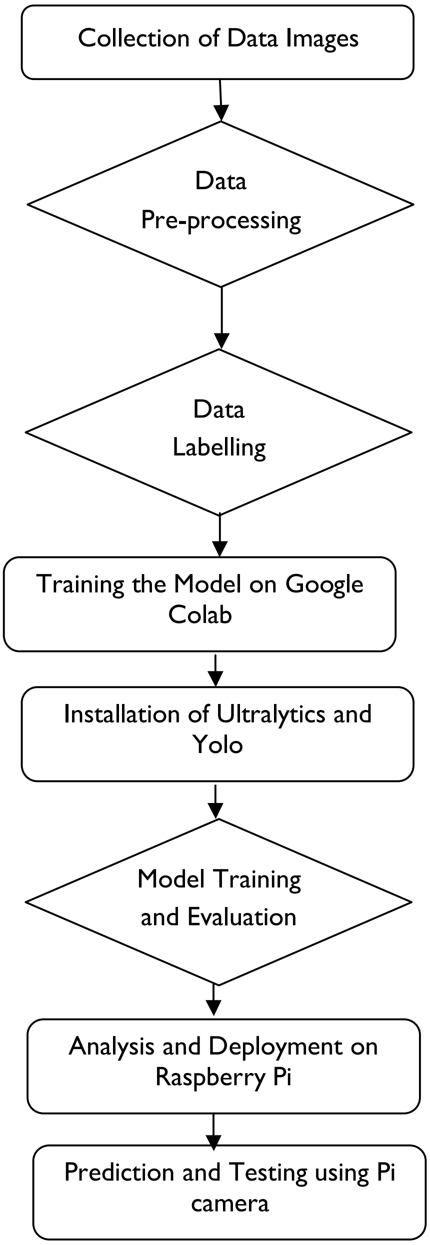

Fig1. Flow chart of methodology

1. Data Collection:

- Car and accident data are collected from various sources, including Google image search and CCTV footage obtained from public roadways or specific locations prone to accidents.
- Google image search queries are tailored to retrieve diverse images representing different types of vehicles and potential accident scenarios.
- CCTV footage is acquired from publicly available sources or through partnerships with relevant authorities or organizations.

2. Data Preprocessing:
- The collected data undergoes preprocessing to ensure consistency and quality.
- Image cropping, resizing, and format standardization are performed to maintain uniformity across the dataset.
- Duplicate images and irrelevant content are removed to streamline the dataset and enhance its relevance to the research objectives.

3. Data Labeling:
- The labeled Img software is employed for annotating the collected images with bounding boxes and class labels.
- Annotators manually label images to indicate the presence of cars and annotate accident scenes, specifying the extent and location of vehicles involved.
- The labeling process ensures the availability of accurately annotated ground truth data essential for training and evaluating the deep learning model.

4. Environment Setup and Dependency Installation:
- The project environment is set up with the necessary software dependencies, including Python, CUDA, and PyTorch.
- Required libraries such as OpenCV, NumPy, and other supporting packages are installed to facilitate data preprocessing, model training, and evaluation.
- Compatibility checks and version management are conducted to ensure seamless integration of components.

5. Model Training Using Ultralytics and YOLOv8:
- The Ultralytics framework is utilized for training the YOLOv8 model on the annotated dataset.
- Training parameters such as learning rate, batch size, and number of epochs are configured based on experimentation and optimization strategies.
- The model undergoes iterative training cycles, where it learns to detect cars and identify accident scenes from the annotated images.
- Transfer learning techniques may be employed to fine-tune pre-trained models on the specific task of car accident detection, leveraging features learned from generic object detection tasks.

6. Model Evaluation:
- The trained model's performance is evaluated using various metrics, including precision, recall, F1-score, and mean Average Precision (mAP).
- Confusion matrices are generated to analyze the model's ability to correctly classify true positives, false positives, true negatives, and false negatives.
- Validation datasets are used to assess the generalization ability of the model and identify potential overfitting or underfitting issues.

7. Predictive Analysis:
- The trained model is deployed to predict car accidents on test data, including real-world images or video streams.
- Predictions are made on unseen data to assess the model's real-time performance and its ability to generalize to new environments and scenarios.
- Prediction results are analyzed to determine the model's accuracy, speed, and robustness in detecting car accidents under varying conditions.

8. Fine-tuning and Iterative Improvement:
- Based on the evaluation results, the model may undergo further fine-tuning and optimization to address any identified shortcomings.
- Hyperparameter tuning, data augmentation, and architectural modifications are explored to enhance the model's performance and reliability.
- Iterative refinement cycles are conducted to iteratively improve the model's effectiveness in car accident detection and minimize false positives and negatives.

9. Predictive Analysis and Deployment on Raspberry Pi:
- The trained model is downloaded and deployed on Raspberry Pi OS, specifically on a Raspberry Pi 3.
- A camera module is attached to the Raspberry Pi to capture live video feeds.
- The program is executed using Thonny IDE, where real-time predictions are made on the captured video stream.
- If an accident is detected, Twilio API is utilized to send alerts to nearby hospitals and police stations, providing essential information about the accident location and severity.
- Additionally, a buzzer is integrated into the system to emit an audible alert when an accident is detected, providing an additional notification mechanism.

10. Comprehensive Testing and Validation:
- The deployed system undergoes comprehensive testing to validate its functionality and performance in real-world scenarios.
- Test data, including live video feeds and simulated accident scenarios, are used to evaluate the system's accuracy, speed, and robustness.
- The effectiveness of the alerting mechanism and the responsiveness of emergency services are assessed through simulated accident detection scenarios.

11. Ethical Considerations:

- Ethical considerations, including privacy concerns related to data collection from CCTV footage and adherence to data protection regulations, are carefully addressed throughout the research process.
- Measures are taken to anonymize and protect sensitive information, ensuring compliance with ethical guidelines and legal requirements

III. METHODOLOGY

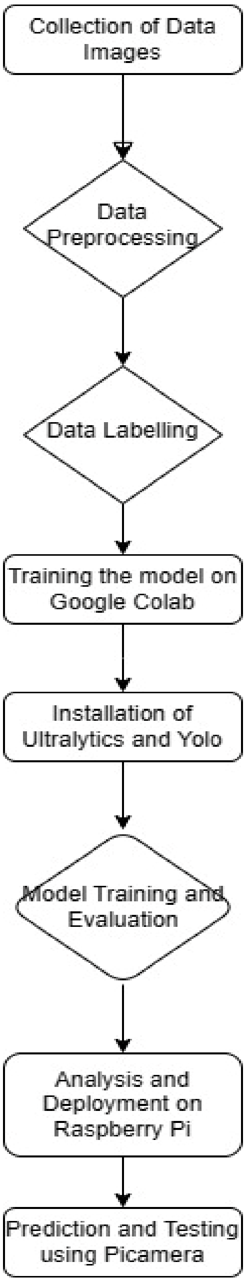

Fig2. Flow chart of methodology

1. Data Collection:

- Car and accident data are collected from various sources, including Google image search and CCTV footage obtained from public roadways or specific locations prone to accidents.
- Google image search queries are tailored to retrieve diverse images representing different types of vehicles and potential accident scenarios.
- CCTV footage is acquired from publicly available sources or through partnerships with relevant authorities or organizations.

2. Data Preprocessing:

- The collected data undergoes preprocessing to ensure consistency and quality.
- Image cropping, resizing, and format standardization are performed to maintain uniformity across the dataset.
- Duplicate images and irrelevant content are removed to streamline the dataset and enhance its relevance to the research objectives.

3. Data Labeling:

- The labeled Img software is employed for annotating the collected images with bounding boxes and class labels.
- Annotators manually label images to indicate the presence of cars and annotate accident scenes, specifying the extent and location of vehicles involved.
- The labeling process ensures the availability of accurately annotated ground truth data essential for training and evaluating the deep learning model.

4. Environment Setup and Dependency Installation:

- The project environment is set up with the necessary software dependencies, including Python, CUDA, and PyTorch.
- Required libraries such as OpenCV, NumPy, and other supporting packages are installed to facilitate data preprocessing, model training, and evaluation.
- Compatibility checks and version management are conducted to ensure seamless integration of components.

5. Model Training Using Ultralytics and YOLOv8:

- The Ultralytics framework is utilized for training the YOLOv8 model on the annotated dataset.
- Training parameters such as learning rate, batch size, and number of epochs are configured based on experimentation and optimization strategies.
- The model undergoes iterative training cycles, where it learns to detect cars and identify accident scenes from the annotated images.
- Transfer learning techniques may be employed to fine-tune pre-trained models on the specific task of car accident detection, leveraging features learned from generic object detection tasks.

6. Model Evaluation:

- The trained model's performance is evaluated using various metrics, including precision, recall, F1-score, and mean Average Precision (mAP).
- Confusion matrices are generated to analyze the model's ability to correctly classify true positives, false positives, true negatives, and false negatives.
- Validation datasets are used to assess the generalization ability of the model and identify potential overfitting or underfitting issues.

7. Predictive Analysis:

- The trained model is deployed to predict car accidents on test data, including real-world images or video streams.
- Predictions are made on unseen data to assess the model's real-time performance and its ability to generalize to new environments and scenarios.
- Prediction results are analyzed to determine the model's accuracy, speed, and robustness in detecting car accidents under varying conditions.

8. Fine-tuning and Iterative Improvement:

- Based on the evaluation results, the model may undergo further fine-tuning and optimization to address any identified shortcomings.
- Hyperparameter tuning, data augmentation, and architectural modifications are explored to enhance the model's performance and reliability.
- Iterative refinement cycles are conducted to iteratively improve the model's effectiveness in car accident detection and minimize false positives and negatives.

9. Predictive Analysis and Deployment on Raspberry Pi:

- The trained model is downloaded and deployed on Raspberry Pi OS, specifically on a Raspberry Pi 3.
- A camera module is attached to the Raspberry Pi to capture live video feeds.
- The program is executed using Thonny IDE, where real-time predictions are made on the captured video stream.
- If an accident is detected, Twilio API is utilized to send alerts to nearby hospitals and police stations, providing essential information about the accident location and severity.
- Additionally, a buzzer is integrated into the system to emit an audible alert when an accident is detected, providing an additional notification mechanism.

10. Comprehensive Testing and Validation:

- The deployed system undergoes comprehensive testing to validate its functionality and performance in real-world scenarios.
- Test data, including live video feeds and simulated accident scenarios, are used to evaluate the system's accuracy, speed, and robustness.
- The effectiveness of the alerting mechanism and the responsiveness of emergency services are assessed through simulated accident detection scenarios.

11. Ethical Considerations:

- Ethical considerations, including privacy concerns related to data collection from CCTV footage and adherence to data protection regulations, are carefully addressed throughout the research process.
- Measures are taken to anonymize and protect sensitive information, ensuring compliance with ethical guidelines and legal requirements.

IV. SYSTEM ARCHITECTURE

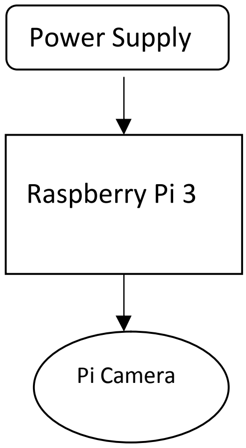

Fig 3. System Architecture

The system architecture for the Car Accident Detection and Alerting System (CADAS) utilizing only Raspberry Pi 3, Pi Camera, and a power supply is designed to be compact, efficient, and capable of real-time accident detection and alerting in resource-constrained environments. Here's a breakdown of the system components and their functionalities:

Raspberry Pi 3:

- The Raspberry Pi 3 serves as the central processing unit for the CADAS, providing computational power for image processing, deep learning inference, and alerting mechanisms.
- It runs the operating system (e.g., Raspberry Pi OS) and hosts the CADAS software stack responsible for managing the entire accident detection and alerting process.

Pi Camera:

- The Pi Camera module is used for capturing live video feeds of the road environment or specific areas where accident detection is required.
- It provides high-resolution images and video streams that are processed in real-time by the CADAS software for accident detection and analysis.

Power Supply:

- A reliable power supply is essential to ensure continuous operation of the CADAS system. This can be achieved using a standard power adapter connected to the Raspberry Pi.
- Battery backup solutions may also be considered for scenarios where a consistent power source is not readily available, ensuring uninterrupted operation of the system.

System Workflow:

- The Pi Camera continuously captures live video feeds of the road environment or designated areas.
- The captured video stream is processed in real-time by the CADAS software running on the Raspberry Pi 3.
- The CADAS software implements the trained deep learning model (e.g., YOLOv8) for object detection, specifically detecting cars and potential accident scenarios in the video frames.
- Upon detecting a potential accident, the CADAS software triggers an alert mechanism, which may include sending notifications to nearby hospitals, police stations, or designated contacts using the Twilio API.
- Additionally, the CADAS software can activate an audible alert mechanism, such as a buzzer, to notify nearby individuals or passersby about the detected accident.
- The CADAS software continuously monitors the video stream for any subsequent incidents, ensuring timely response and assistance in case of multiple accidents or emergencies.

Key Features:

- Compact and portable architecture suitable for deployment in various environments, including urban roads, highways, and remote areas.
- Real-time accident detection and alerting capabilities, enabling prompt response and assistance to accident victims.
- Low-cost solution leveraging off-the-shelf components, making it accessible and scalable for deployment in diverse settings.
- Minimal hardware requirements, making it suitable for resource-constrained environments where power and computational resources may be limited.

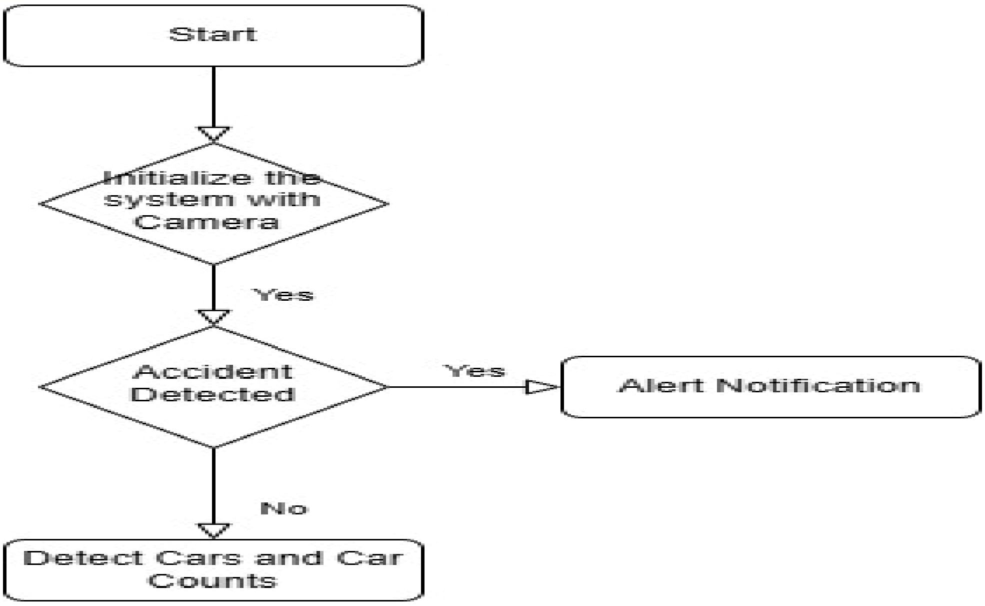

Fig 4. Accident Detection Alert System

V. HARDWARE COMPONENTS

Fig 5. Raspberry Pi3

Raspberry Pi serves as the cornerstone of our project, offering a compact and versatile computing platform ideal for implementing our Car Accident Detection and Alerting System (CADAS). With its small form factor, low power consumption, and robust performance capabilities, Raspberry Pi empowers our system to perform real-time image processing, deep learning inference, and alerting mechanisms using minimal resources. Its integration with the Pi Camera enables seamless capture of live video feeds, while its GPIO pins facilitate the incorporation of additional hardware components such as buzzers for audible alerts. Raspberry Pi's affordability and accessibility make it an ideal choice for deploying CADAS in diverse environments, ranging from urban roads to remote areas, contributing to enhanced road safety and emergency response efforts.

Fig 6. Pi Camera

The PiCamera module, tailored for Raspberry Pi, serves as the primary sensor in our Car Accident Detection and Alerting System (CADAS), delivering high-resolution images and real-time video feeds. Its compact design and seamless integration enable effortless deployment, while its advanced imaging capabilities provide crisp and detailed imagery essential for accurate object detection. With adjustable parameters for resolution and exposure, the PiCamera ensures optimal performance in various lighting conditions. Its low latency and hardware-accelerated encoding support facilitate rapid processing of video streams, enabling timely identification of potential accident scenarios. Overall, the PiCamera module plays a crucial role in CADAS, contributing to the development of a robust and efficient solution for enhancing road safety and emergency response efforts.

VI. RESULTS AND DISCUSSION

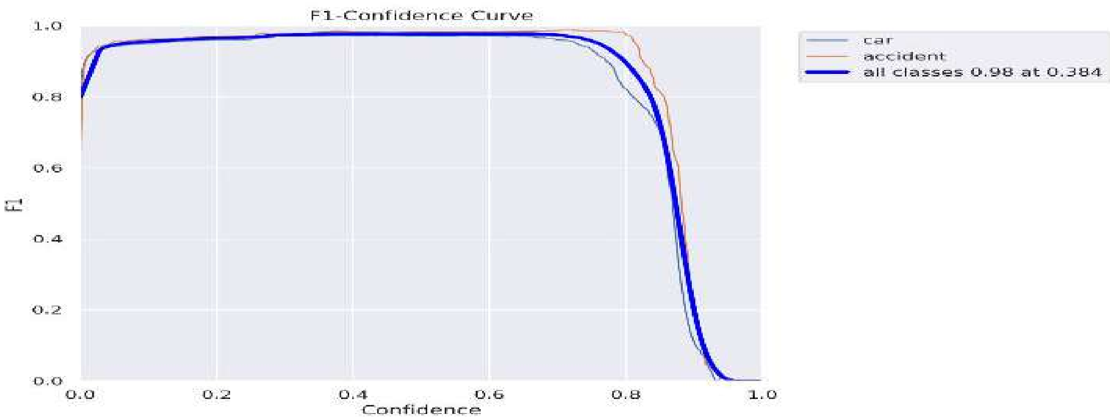

Fig 7. F1 Confidence Curve

In the context of your project, the F1 confidence curve serves as a valuable tool for evaluating the performance of the Car Accident Detection and Alerting System (CADAS) in detecting and classifying car accidents. The F1 confidence curve depicts the relationship between the confidence threshold (or decision threshold) used for classifying detections as positive or negative and the corresponding F1 score. At lower confidence thresholds, the CADAS may classify more detections as positive, including true positives and false positives. As the confidence threshold increases, the CADAS becomes more conservative in its classifications, resulting in fewer positive detections but potentially higher precision. The F1 confidence curve provides insights into the trade-off between precision and recall at different confidence thresholds. A higher F1 score indicates a better balance between precision and recall, reflecting the CADAS's ability to accurately detect car accidents while minimizing false alarms. Analyzing the F1 confidence curve allows us to identify the optimal confidence threshold that maximizes the CADAS's performance for our specific application. By selecting an appropriate threshold, we can fine-tune the CADAS to meet operational requirements, such as minimizing false alarms while maintaining high detection accuracy. Moreover, the F1 confidence curve enables us to assess the CADAS's robustness across varying confidence thresholds, providing valuable insights into its reliability and performance under different operating conditions. This analysis helps in making informed decisions regarding system configuration and deployment strategies to optimize CADAS's effectiveness in enhancing road safety and emergency response efforts.

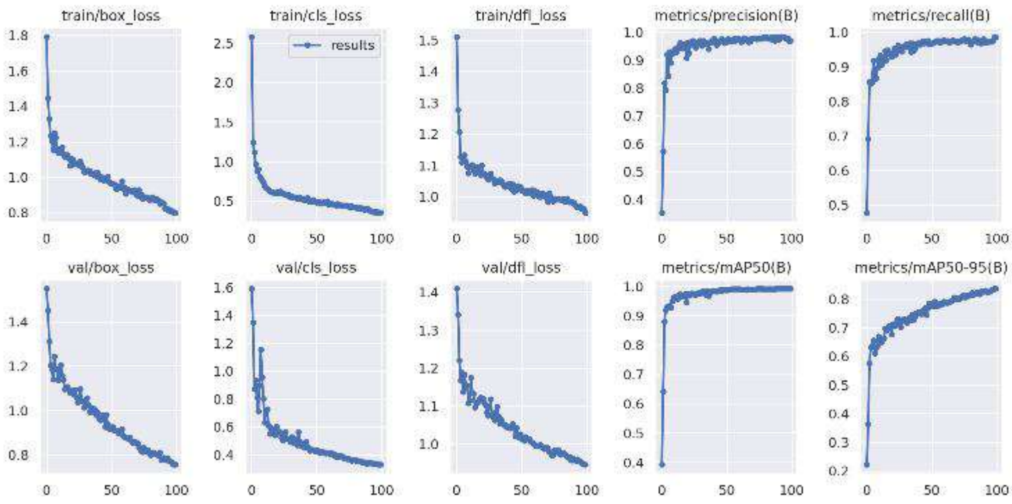

Fig 8. Train and Validation Loss

In the context of your project, the Train and Validation Loss Graph serves as a critical visualization tool for assessing the performance of your Car Accident Detection and Alerting System (CADAS) during the training process. This graph illustrates the evolution of both the training loss and the validation loss over successive iterations or epochs of model training. As the CADAS model learns from the training data, the training loss reflects how well the model is fitting the training data. A decreasing trend in the training loss indicates that the model is improving its ability to minimize errors and accurately classify car accidents during training. Concurrently, the validation loss provides insight into the model's generalization ability by evaluating its performance on unseen validation data. A decreasing trend in the validation loss indicates that the model is effectively generalizing its learned patterns to new, unseen data. The Train and Validation Loss Graph allows us to monitor the convergence of the CADAS model during training. Ideally, both the training loss and the validation loss should decrease steadily over successive epochs, indicating that the model is learning relevant features from the training data without overfitting. A widening gap between the training loss and the validation loss may indicate overfitting, where the model performs well on the training data but fails to generalize to new data. Conversely, if the validation loss starts to increase while the training loss continues to decrease, it may indicate that the model is overfitting and failing to generalize effectively. By analyzing the Train and Validation Loss Graph, we can make informed decisions about model architecture, hyperparameters, and training strategies to optimize CADAS's performance. Fine-tuning these aspects based on insights from the loss graph can help improve the model's accuracy and reliability in detecting car accidents, ultimately enhancing road safety and emergency response efforts.

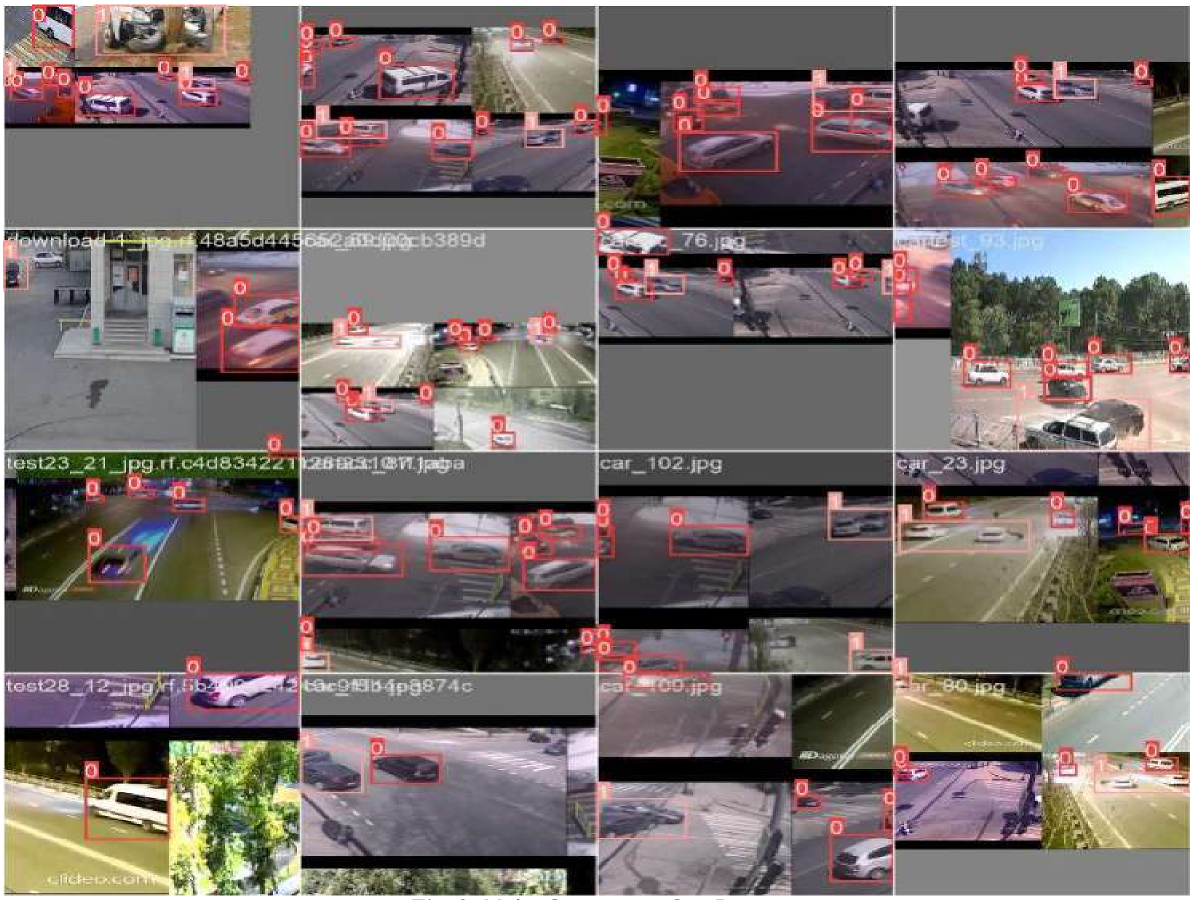

Fig 9. Yolo Output on Car Detection

The output of YOLO (You Only Look Once) on car detection and accident detection is characterized by its ability to accurately identify and localize vehicles and potential accident scenarios in images or video streams. YOLO outputs bounding boxes around detected objects, including cars and other relevant elements, providing precise spatial information about their location within the scene. Additionally, YOLO annotates these bounding boxes with text labels, indicating the class of the detected object (e.g., "car" or "accident"), along with confidence scores representing the model's confidence in its predictions. This comprehensive output format enables seamless interpretation and analysis of detected objects, facilitating real-time decision-making in applications such as road safety and accident prevention. Furthermore, YOLO's output often includes confidence scores associated with each detected object, offering insights into the model's confidence level in the accuracy of its predictions. This allows for thresholding based on confidence scores, enabling the system to prioritize highly confident detections while filtering out potentially spurious results. Additionally, YOLO's efficient single-pass architecture ensures fast inference speeds, making it well-suited for real-time applications such as car and accident detection. Overall, YOLO's comprehensive output format, coupled with its speed and accuracy, makes it a powerful tool for enhancing safety measures and response mechanisms in various environments**.**

Fig 10. Yolo output on prediction

YOLO's output on prediction is marked by its efficiency in detecting and classifying objects within images or video frames in real-time. It provides a structured format that includes bounding boxes drawn around detected objects, along with corresponding labels indicating the object classes, such as "car" or "accident." Additionally, YOLO assigns confidence scores to each prediction, representing the model's certainty in its classifications. This comprehensive output allows for rapid interpretation and decision-making, making YOLO an invaluable tool for various applications, including car accident detection, where timely responses are critical for ensuring safety on the roads.

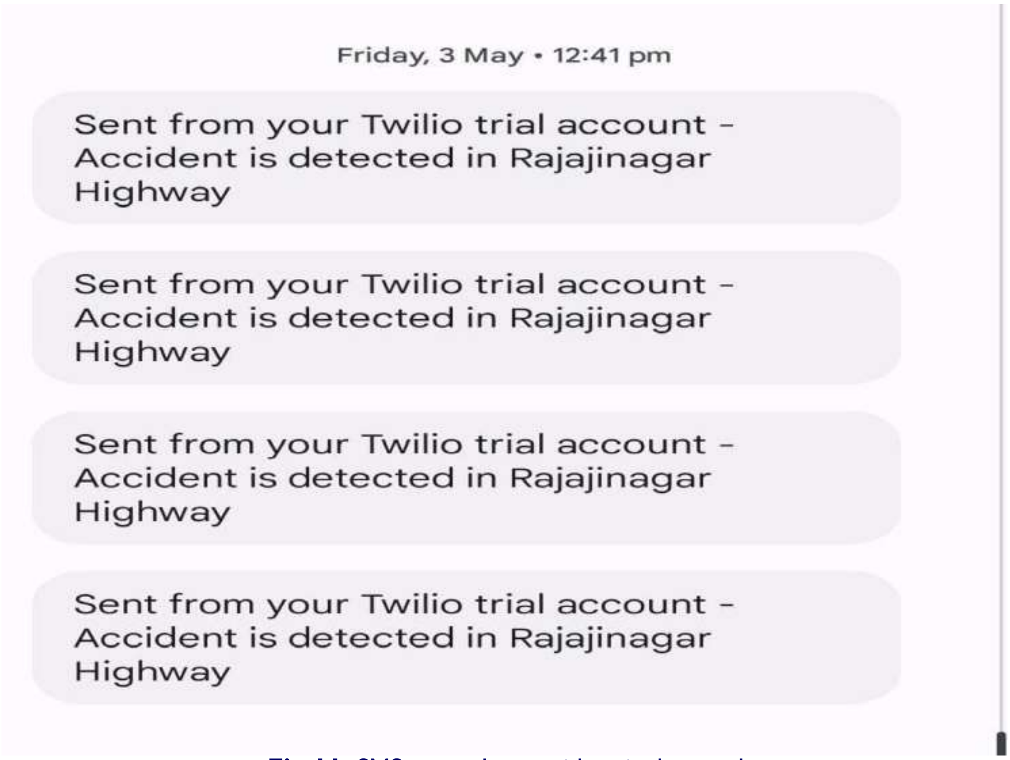

Fig 11. SMS sent when accident is detected

The alerting system using Twilio in response to detected accidents enhances the Car Accident Detection and Alerting System (CADAS) by providing timely notifications to relevant stakeholders. Upon detecting a car accident through the CADAS, the system triggers an alert mechanism that utilizes Twilio's messaging capabilities to send notifications to designated recipients, such as nearby hospitals, police stations, or emergency contacts. Twilio enables the CADAS to deliver real-time alerts via SMS, voice calls, or other communication channels, ensuring that crucial information about the accident, including its location and severity, is promptly communicated to the appropriate authorities and individuals. This immediate notification facilitates rapid response and assistance, potentially reducing the response time and minimizing the impact of the accident. Moreover, Twilio's flexible API allows for customization of alert messages, enabling the CADAS to include relevant details about the accident, such as the number of vehicles involved, potential injuries, and any additional information deemed necessary for efficient emergency response. By leveraging Twilio's robust communication platform, the CADAS enhances its effectiveness in alerting relevant stakeholders about detected accidents, contributing to improved road safety and emergency response efforts.

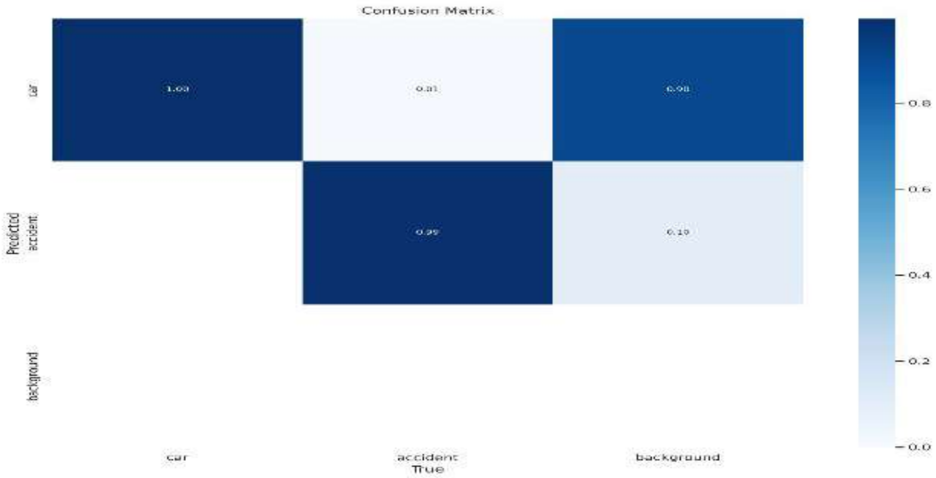

Fig 12. Confusion Matrix

The confusion matrix serves as a critical tool for evaluating the performance of the Car Accident Detection and Alerting System (CADAS) by providing insights into the model's classification accuracy and error types. In the context of CADAS, the confusion matrix categorizes predictions made by the model into four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). True positives (TP) represent instances where CADAS correctly detects and alerts about car accidents. These are scenarios where the model accurately identifies the presence of an accident, leading to a timely response and potential assistance. False positives (FP) occur when CADAS incorrectly identifies a non-accident scenario as an accident. While false positives may lead to unnecessary alerts, they can be minimized through fine-tuning the model's threshold parameters and improving its specificity. True negatives (TN) denote instances where CADAS correctly identifies the absence of a car accident. These are situations where the model accurately recognizes non-accident scenarios, contributing to the system's overall reliability and trustworthiness. False negatives (FN) represent instances where CADAS fails to detect an actual car accident. False negatives are particularly concerning as they may result in delayed or missed alerts, potentially compromising emergency response efforts. By analyzing the values in the confusion matrix, we can calculate various performance metrics such as precision, recall, and F1-score, which provide a comprehensive assessment of CADAS's effectiveness in car accident detection. Additionally, the confusion matrix enables us to identify specific areas for improvement, such as reducing false positives or minimizing false negatives, to enhance the system's overall performance and reliability.

VII. CONCLUSION

In conclusion, our Car Accident Detection and Alerting System (CADAS) represent a significant advancement in leveraging deep learning and real-time communication technologies to enhance road safety and emergency response efforts. Through the integration of state-of-the-art object detection algorithms like YOLOv8 and the utilization of Twilio for instant alerting, CADAS demonstrates its capability to accurately detect car accidents and promptly notify relevant authorities and stakeholders. However, while our project has shown promising results, there are avenues for further improvement. One such area is the enhancement of model performance through the acquisition of more diverse and extensive datasets. Increasing the dataset size can help improve the model's accuracy and generalization ability, especially in detecting rare or unusual accident scenarios. Additionally, focusing on optimizing the computational efficiency of the CADAS, particularly on Raspberry Pi, is crucial for achieving faster frame rates and real-time performance. Techniques such as model compression, hardware acceleration, and algorithmic optimizations can be explored to maximize the system's computational efficiency while maintaining accuracy. Furthermore, integrating additional sensors or data sources, such as GPS or accelerometer data, could enrich the CADAS's capabilities, enabling more accurate accident localization and severity assessment. In summary, while our project has laid a solid foundation for car accident detection and alerting, there is still ample room for refinement and expansion. By addressing these areas of improvement, we can further enhance the effectiveness and reliability of CADAS, ultimately contributing to safer roads and more efficient emergency response systems.

REFERENCES

[1]. Nedjet Dogru, Abdulhamit Subasi, "Traffic Accident Detection By Using Machine Learning Methods", International Symposium on Sustainable Development , March 2012
[2]. Vipul Rana, Hemant Joshi, Deepak Parmar, Pradnya Jadhav Monika Kanojiya, "Accident Detection using Machine Learning Algorithm", International Research Journal of Engineering and Technology(IRJET), March 3, 2019
[3]. Deeksha Gour , Amit Kanskar, "Optimised YOLO: Algorithm for CPU to Detect Road Traffic Accident and Alert System", International Journal of Engineering Research & Technology (IJERT), September 16, 2019
[4]. Xi Jianfeng, Guo Hongyu, Tian Jian, Lisa Liu, Liu Haizhu, "A Classification and Recognition Model for the Severity of Road Traffic Accidents ", SAGE Journal , May 22 ,2019

[5]. Raad Ahmed Hadi, Ghazali Sulong, Loay Edwar George, "Vehicle Detection and Tracking Techniques: A Concise Review", Signal and Image Processing: an International Journal, Vol 5, February, 2014
[6]. Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, "Object Detection with Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, April 16, 2019
[7]. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, "Mask R-CNN", Facebook AI Research (FAIR), 2018
[8]. Sodikov, Jamshid, "Road Traffic Accident Data Analysis and Visualization in R", International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR), 2018
[9]. S.Ramya, SKReshma, V. Dhatri Manogana, Y. Satya Saroja, "Accident Severity Prediction using Data Mining Methods", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol 5, Issue 2, 2019
[10]. Jabar H. Yousif, Al Rababba Mamoun S, "Neural Technique for Predicting Traffic Accidents in Jordan", Journal of American Sciences, November, 2013