International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

Real-Time Vehicle Accident Recognition from Traffic Video Surveillance using YOLOV8 and OpenCV

Deepak T. Mane[1*], Sunil Sangve[2], Sahil Kandhare[3], Saurabh Mohole[4], Sanket Sonar[5], Satej Tupare[6]
[1]Vishwakarma Institute of Technology, Pune-411037, Maharashtra, India
[1]dtmane@gmail.com
[2,3,4,5,6]JSPM's Rajarshi Shahu College of Engineering, Pune-411033, Maharashtra, India
[2]sunilsangve@gmail.com
[3]sahilkandhare07@gmail.com,
[4]saurabhmohole@gmail.com
[5]sanketsonar2002@gmail.com
[6]satejtupare@gmail.com

Abstract—The automatic detection of traffic accidents is a significant topic in traffic monitoring systems. It can reduce irresponsible driving behavior, improve emergency response, improve traffic management, and encourage safer driving practices. Computer vision can be a promising technique for automatic accident detection because it provides a reliable, automated, and speedy accident detection system that can improve emergency response times and ultimately save lives. This paper proposed an ensemble model that uses the YOLOv8 approach for efficient and precise event detection. The model framework's robustness is evaluated using YouTube video sequences with various lighting circumstances. The proposed model has been trained using the open-source dataset Crash Car Detection Dataset, and its produced precision, recall, and mAP are 93.8% and 98%, 96.1%, respectively, which is a significant improvement above the prior precision, recall, and mAP figures of 91.3%, 87.6%, and 93.8%. The effectiveness of the proposed approach in real-time traffic surveillance applications is proved by experimental results using actual traffic video data.

Keywords-Anomal Detection; Accident detection; Computer vision; deep learning; YOLOV8

I. INTRODUCTION

All Road traffic accidents have been a significant public safety concern worldwide. The World Health Organization (WHO) estimates those road traffic accidents cause 50 million extra injuries yearly and 1.35 million fatalities. As a result, there is an increasing demand for dependable, reliable, and efficient systems capable of detecting road accidents and responding quickly. Computer vision has developed as a viable tool for road accident detection and response, capturing real-time traffic data and analyzing it for signs of accidents using cameras and sensors. You Only Look Once version 8 (YOLOv8) and OpenCV are widely used tools for object detection and image processing among the computer vision algorithms available. This paper aims to use YOLOv8 and OpenCV to create an accurate and efficient system for the real-time recognition of car crashes in traffic surveillance. The proposed approach can help with rapid accident response, improve emergency response, and reduce the impact of accidents on public safety. There are numerous and diverse applications for accident detection using computer vision, potentially improving traffic management, transportation safety, emergency response, insurance, Autonomous Vehicles, Smart Cities, and many other fields. Accident detection systems based on computer vision have the potential to save lives and improve public safety.

- Overall, this research paper contributes to are
- To the development of computer vision-based systems for road accident detection and response. The proposed system, which uses YOLOv8 and OpenCV, can enhance emergency response, reduce the adverse effects of accidents on public safety, and even save lives.
- The results of this study can be used to create more effective and efficient methods for detecting and responding to traffic accidents, which would eventually increase public safety and help save lives.

The paper begins by examining existing computer vision-based road accident detection research, emphasizing YOLOv8 and the OpenCV. The literature survey covers the fundamentals of computer vision and the numerous methodologies that have been utilized in the past, represented in section II. The proposed YOLO8 and OPenCV8 ensemble model is described in section III. The Proposed Architecture section of the paper

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

outlines the steps involved in training and testing the YOLOv8 and OpenCV algorithms using real-world traffic surveillance footage. The methodology includes model training, model evaluation, and data gathering. The OpenCV method was used to classify the images and establish whether a crash occurred. In contrast, the YOLOv8 algorithm was used to recognize automobiles' presence and position in the frame. The accuracy of the proposed system was evaluated using various performance metrics, including precision, recall, mAP, and F1 score. In section IV, we discussed the mathematical model behind the YOLO architecture. Section V presents the experiments carried out to gauge the proposed system's effectiveness. The results of the experiments demonstrate that the proposed method was highly accurate in real-time detecting and identifying car crashes. The system's accuracy was better than other approaches, demonstrating the value of the suggested strategy. The conclusion is represented in section VII.

II. LITERATURE SURVEY

There have been breakthroughs in the field of real-time recognition of vehicle crashes in traffic monitoring over the last decade. Numerous researchers have attempted to enhance detecting systems' efficiency and precision. Deep learning techniques such as convolutional neural networks (CNNs) have been widely used in recent years for object detection in real-time video streams.One of the most difficult challenges in this area has been to obtain effective processing while retaining high accuracy in crash detection. Numerous researchers have proposed several approaches for accomplishing this, including the use of region-based CNNs (R-CNNs) and the YOLO (You Only Look Once) algorithm. Researchers have also investigated the use of audio and vibration sensors, in addition to visual sensors, to improve detection performance. Machine learning techniques have been employed in research to analyze sensor data and increase crash detection accuracy.

2.1 Vehicle crash detection based on machine learning Techniques:

Vehicle accident detection using machine learning approaches involves applying a variety of models and algorithms to detect crashes involving cars in real-time. The major purpose of these strategies is to enable early detection and response to car accidents, which can assist save lives and reduce injury severity.

Y. Wang et al.[1] (2020) give a thorough analysis of many machine learning methods used for accident detection and management. The authors stated that machine-learning algorithms for vehicle collision detection and management have shown promising results and that they can dramatically reduce traffic congestion and increase road safety.

The paper examines the most often used machine learning methods for identifying and managing traffic incidents, such as rule-based systems, decision trees, neural networks, support vector machines, and deep learning.

2.2 Vehicle crash detection based on the Yolo Algorithm:

N. C. Suriya et al.[2] (2020) offer a full review of the You Only Look Once (YOLO) algorithm and its application in traffic accident detection and analysis in their paper published in 2020. The YOLO algorithm is found to be a popular and successful way of detecting traffic accidents in real-time, according to the research. It may be used to accurately analyze traffic flow and spot accidents, making it an invaluable tool for traffic management and emergency services.

2.3 Literature Review (2014- 2022):

According to the referred three publications (2014) research by M. I. Taha and A. Almohaimeed [3], K. Lu, D. D. Lin [4], A. L. Nogueira and M. M. Oliveira [5] the study proposes three real-time techniques for CCTV or on-road video camera footage-based autonomous traffic accident detection. They achieve high accuracy rates in tests conducted in the real world by using machine learning algorithms and computer vision techniques to identify various forms of events, such as collisions and stopped automobiles. The technologies can update drivers on the status of traffic in real-time and send notifications to traffic management authorities. As per A. Mathur et al. (2015) research[6] the use of image processing techniques such as background reductions and motion tracking, as well as machine learning algorithms, is effective in detecting accidents. M. Rizwan et al.[7] (2016) proposed an image processing-based system for real-time car accident detection. Using edge detection, segmentation, and object tracking techniques, the system correctly detected accidents. In the year 2019, the research was done in a few papers involving papers by author Muhammad Rizwan et al. [8], S. M. Sabrin et al. [9], Jihong Wang et al. [10], and J. Lee et al. [11] have all conducted research in the field of Real-time Vehicle Crash Detection Using ML, based on various techniques such as Deep Learning Techniques and CNN. They discovered ML algorithms are robust for completing the task of vehicle detection, as well as for improving traffic systems around the world. L. Zhang et al.[12] (2020) presented a paper in which they used deep learning-based object detection methods for traffic surveillance, which can help improve traffic security as well as effectiveness. In addition, the research presents a novel multi-scale feature fusion strategy to improve detection efficiency while decreasing false positives. The YOLO algorithm was used for Traffic Monitoring Systems in four

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

publications published in 2021 by authors M. Saleem et al.[13], Z. Yang et al.[14], Sun, J. et al.[15], and T. D. Nguyen et al.[16]. They have achieved better results. The research of M. Saleem et al. is based on YOLOv3 and machine learning. Researchers have also identified the need to improve the system's effectiveness in low-light circumstances and to expand its coverage to additional regions.[13]. In 2022, Mane D. T. represented the compressive survey on anomaly detection using machine learning [18].

2.4 Why should we use YOLO for object detection?

While both ML techniques and YOLO can be used to detect car crashes, YOLO has some major advantages, such as high accuracy, speed, and flexibility, making it a popular choice for real-time object identification applications. Nonetheless, in certain instances, ML algorithms can still be successful at identifying crashes, particularly when a high number of labeled images are available for training which is represented in Table1. The key difference between the two approaches is the way they process the data. Overall, ML algorithms and YOLO can both be used to identify car crashes, and which one to use will rely on the application's particular needs, such as accuracy, speed, and resource limitations.

TABLLE 1. Comparison of vehicle crash detection algorithms based on machine learning (ML) and YOLO

Criteria	**Vehicle crash detection using ML algorithms**	**Vehicle crash detection using YOLO**
Detection Accuracy	The detection accuracy varies depending on the method and dataset used. To attain high accuracy, a significant number of manually labeled images may be required.	Achieves cutting-edge accuracy on a variety of object detection benchmarks, including the COCO and VOC datasets.
Speed	The detection speed can vary based on the algorithm and technology used. To attain real-time performance, a large amount of computer power may be required.	YOLO is designed for real-time detection and can accomplish fast detection even on low-end hardware.
Object Size	Little or distant objects and even objects with poor contrast or obstructed objects, may be difficult to identify.	YOLO is designed to detect objects of all sizes and scales, including small objects, and can efficiently manage occlusions and cluttered scenes.
Training Time	Deep learning models may necessitate lengthy training time for ML algorithms.	Because of its simplicity and streamlined architecture, YOLO is faster to train than many other object detection algorithms.
Dataset Size	To attain high accuracy, ML algorithms may require a large labeled dataset.	YOLO is commonly trained on a subset of bigger datasets and can achieve excellent accuracy with smaller datasets.
Flexibility	Significant customization may be required to recognize specific sorts of items or to adapt to specific use situations.	YOLO is extremely adaptable and may be easily applied to a wide range of applications, including traffic monitoring and car crash detection.
Ease of Use	To build up and use ML algorithms efficiently, extensive technical expertise may be required.	YOLO is pretty simple to use, with pre-trained models and simple APIs.

III. PROPOSED ARCHITECTURE

By sending alerts to authorities when a vehicle crash is detected, the proposed model technology will assist in creating an intelligent transportation system that will enable us to reduce the number of fatalities brought on by traffic accidents. Object detection and crash identification are the two primary modules of the proposed ensemble system for the real-time identification of car crashes in traffic surveillance using YOLOv8 and OpenCV, which is represented in Figure 1. Previous work in the field used YOLOv5 to get the results mentioned. However, in our proposed ensemble model, we used YOLOv8. As a result, we produced improved results. Consequently, according to our literature review, both YOLOv8 and YOLOv5 have advantages and disadvantages when selecting the best object detection model. Although YOLOv5 is more user-friendly, YOLOv8 is faster and more accurate. For applications that require real-time object detection, YOLOv8 is the ideal choice.

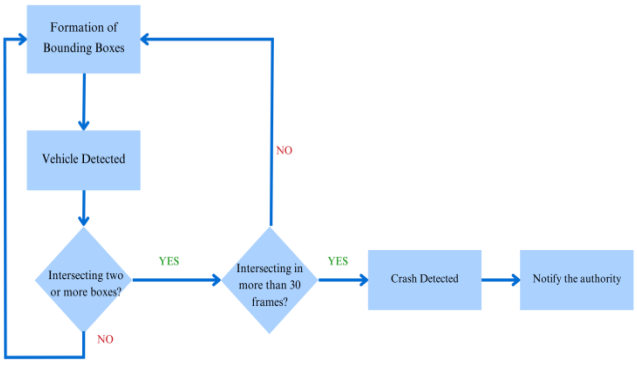

FIGURE 1: Flowchart of the proposed system.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

A. Module 1 Object Detection:

The first part of the proposed system is object detection, which scans the traffic surveillance footage for the presence of cars using YOLOv8. A significant dataset of car picture training data is used to train the YOLOv8 algorithm. The system is capable of properly locating and tracking the cars in real-time, supplying crucial data for the following steps

YOLOv8 for object detection: A popular deep learning technique for object detection in real-time applications is YOLOv8 (You Only Look Once version 8). It features several modifications to increase accuracy and performance over its previous version, YOLOv7

The input image is first split up into a grid of cells, and then the YOLOv8 algorithm forecasts a set of bounding boxes and their related class probabilities. The algorithm extracts feature maps from the input image using a pre-trained convolutional neural network (CNN) architecture. To detect objects, it then employs a sequence of convolutional layers that combine both local and global information about the image

The network produces a 3D tensor as its output, with the first two dimensions denoting grid cells and the third denoting projected class probabilities and bounding boxes. Each bounding box is represented by four values: the height and width of the box, its center coordinates, and a confidence score that expresses how likely it is that the object will be found inside the box. The class probabilities predict the likelihood that each object in the dataset belongs to each class. To eliminate redundant detections, YOLOv8 employs non-maximum suppression after creating the collection of bounding boxes and class probabilities. This procedure comprises deleting boxes that significantly overlap with higher-scoring boxes and sorting the bounding boxes according to their confidence scores. The result is a list of the final detected items along with their corresponding bounding boxes and class probabilities. This phase makes sure that each object is only detected once

B. Stepwise learning steps for YOLOv8 for detecting objects as follows

- *Step 1*: Load an input image
- *Step 2:* Resize the input image to the desired input size of the YOLO model
- *Step 3:* Normalize the input image pixel values to a range of 0 to 1
- *Step 4*: Pass the normalized input image through the YOLO model
- *Step 5:* Obtain the predicted bounding boxes, confidence scores, and class probabilities for the detected objects
- *Step 6:* Apply a threshold to the confidence scores to select the most confident detections
- *Step 7:* Apply non-maximum suppression to remove overlapping bounding boxes for the same object
- *Step 8*: Draw the remaining bounding boxes on the input image and label them with their corresponding object classes
- *Step 9:* Display the output image with the detections

This pseudo-code represents the primary steps in YOLOv8's object detection pipeline. It loads and preprocesses the input image, then runs it through a pre-trained CNN architecture to extract features before applying post-processing techniques (such as non-maximum suppression) to reduce redundant detections. Ultimately, for the recognized items in the image, the algorithm returns a list of bounding boxes, class labels, and confidence ratings

C. Module 2 Crash Identification:

The proposed model second component is crash identification, which uses OpenCV to classify images and determine if a crash has happened. The OpenCV system, which was trained on a large dataset of accident photos, can effectively recognize various sorts of crashes, including side-impact collisions head-on collisions, and rear-end collisions. In the event of a crash, the algorithm evaluates the input data in real-time.

The open-source computer vision and machine learning package OpenCV (Open Source Computer Vision) provides an extensive collection of image and video processing tools. Its learning steps are

- Step 1: Load and read the input image.
- Step 2: Convert the input image to grayscale.
- Step 3: Apply image processing techniques (e.g. blurring, thresholding, edge detection).
- Step 4: Apply feature detection and extraction algorithms (e.g. Hough Transform, SIFT, SURF).
- Step 5: Visualize the output

IV. MATHEMATICAL DESCRIPTION OF PROPOSED MODEL

The YOLO (You Only Look Once) object detection model consists of several mathematical equations that are used to process the input image and predict the bounding boxes and class probabilities for objects in the image. The feature extraction backbone consists of multiple convolutional layers that are used to extract features from the input image. The mathematical equations involved in the feature extraction backbone include.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

A. Convolutional Layer:

Each convolutional layer applies a set of filters to the input image to extract features. The mathematical equation for a 2D convolution operation is:

$$y(i,j) = (w * x)(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} w(u,v)x(i-u, j-v) \quad (1)$$

The output feature map y at point (i,j) is represented by this equation as a weighted sum of the input feature map x with filter coefficients w over a limited region centered at (i,j). The filter is k × k in size, and the indices u and v are added together to span the full filter region. The convolution operation is represented by the * symbol.

B. Max Pooling Layer:

Each max pooling layer downsamples the feature map by retaining the maximum value within each non-overlapping region. The mathematical equation for a max pooling operation is:

$$y(i,j) = \max_{u,v}(x(i+u, j+v)) \quad (2)$$

This equation expresses the largest value in a non-overlapping region of size (u, v) in the input feature map x as the maximum value in the output feature map y at location (i,j). The max function represents the maximum operation, and the indices u and v are added together to cover the full pooling zone

Object Detection Head:

The object detection head consists of several layers that are used to predict the bounding boxes and class probabilities for objects in the input image. The mathematical equations involved in the object detection head include

a. Detection Layer:

The detection layer predicts the class probabilities and bounding box coordinates for objects in the image. The mathematical equation for the detection layer is:

$$\text{bbox}_{i,j} = \text{tx}_{i,j}\ \sigma\ (\text{tw}_{i,j}) + \text{b}_{i,j}$$
$$\text{conf}_{i,j} = \sigma(\text{tc}\ i,j)$$
$$\text{class}_{i,j,c} = \text{pc}_{i,j,c}\ \sigma\ (\text{tc}_{i,j,c}) \quad (3)$$

The predicted bounding box coordinates are represented by bbox, the objectness score is represented by conf, and the class probabilities are represented by a class. tx and t_w are the projected center coordinates and width/height offsets, respectively, while b is the cell's anchor box (i,j). The confidence and class prediction offsets for each anchor box are represented by t_c, and the anticipated probability of the anchor holding the associated object class c is represented by pc. The sigmoid function is denoted by the symbol sigma.

C. Anchor Boxes:

Anchor boxes are pre-defined boxes of different sizes and aspect ratios that are used to predict the bounding box coordinates. The mathematical equation for anchor boxes is:

$$\omega_\alpha = \Pi_{\omega\alpha}\ \varepsilon^{\tau\_\omega}$$
$$\eta_\alpha = \Pi_{\eta\alpha}\ \varepsilon^{\tau\_\eta} \quad (4)$$

Here, w_a and h_a represent the width and height of the anchor box, respectively. p_wa and p_ha are the width and height of the default anchor box, respectively, and t_w and t_h are the projected offsets for width and height. These offsets are anticipated by the neural network during training and are used to alter the size of the default anchor box to better match the size of objects in the image. The exponential function e is employed to verify that the projected values for w_a and h_a are positive.

D. Non-Maximum Suppression (NMS):

Non-maximum suppression is used to eliminate duplicate detections and select the most confident predictions.
The mathematical equation for non-maximum suppression is

$$\text{NM}\Sigma(B, \Sigma, T) = \{\beta_\iota \in B \mid \forall\ \beta_\iota \in B, \iota \neq \varphi: \text{IoY}(\beta_\iota, \beta_\varphi < T) \quad (5)$$

V. Experiment Results

To evaluate the performance of the proposed ensemble model for real-time recognition of vehicle crashes in traffic surveillance, we conducted experiments with the help of YouTube videos containing a variety of traffic circumstances, including congested traffic, accidents, and regular traffic flow. The videos ranged in length from one minute to ten minutes. During testing, we extracted individual frames from films and ran them through the YOLOv8 object detection model in real-time to detect probable crashes. We used numerous evaluation standards to evaluate the performance of the proposed model, including precision, recall, mAP, and F1 score.

FIGURE 2: Working of the proposed system in real-time

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
Article Received: 28 February 2023 Revised: 09 April 2023 Accepted: 25 April 2023

Dataset Used: The Crash Vehicle Detection dataset was generated by the Mada Study team and is hosted on Roboflow Universe [17]. This dataset includes 2525 images, each of which has associated annotations for crashed cars in the form of bounding boxes around the vehicles in the scene. The dataset includes images from multiple sources, including surveillance cameras and social media, it depicts a variety of traffic events, including crashes, congested traffic, and normal traffic flow
The dataset can be used to construct and test object detection models for real-time car crash detection in traffic surveillance. The annotations can be used to train object detection models to identify crashed vehicles more precisely and increase the safety of traffic monitoring systems. The dataset can be downloaded and used for free under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. We used the Crash Car Detection dataset to test the performance of proposed method for the real-time identification of vehicle crashes in traffic surveillance. We were able to test our technique on a variety of traffic circumstances, including crashes and congested traffic, thanks to the dataset. We trained our object detection model using the annotations in the dataset and compared its performance to existing approaches for vehicle crash detection in traffic surveillance

Hardware and Software Used: Google Colab's Runtime GPU is used for model training with a Tesla T4 graphics card is presented in Table 2. The Tesla T4 is a powerful GPU built for data center workloads and is ideal for deep learning model training. The Tesla T4 with Google Colab's Runtime GPU supplied us with the processing capabilities needed for efficient and effective model training. NVIDIA System Management Interface (SMI) is used to monitor and adjust the GPU's performance throughout training to fully utilize the capability of the Tesla T4 graphics card. The NVIDIA GPU software stack also includes the driver version 525.85.12 and CUDA version 12.0, which allow software applications to connect with the GPU hardware and use its computing capabilities. With a 640-pixel picture, we trained our model for 50 epochs. The number of epochs determines how often the model analyses the whole dataset during training, whereas the image size specifies the resolution at which the model processes the input images. Raising the number of epochs and image size may enhance model accuracy, but it may also necessitate more computational resources and result in longer training times. 50 epochs completed in 1.471 hours. Model comparison with previous work represented in Table 3 and its graphical representation of TABLE 3 is represented in Figure 3.

TABLE 2: Hardware and Software Specifications

Tool/Equipment	**Description**
GPU	Tesla T4
GPU Purpose	Model training
Platform	Google Colab's Runtime GPU
GPU Performance Monitoring	NVIDIA System Management Interface (SMI)
GPU Driver Version	525.85.12
CUDA Version	12.0
Advantage	Powerful GPU built for data center workloads

TABLE 3: Model comparison with previous work

Model	**Precision**	**Recall**	**mAP**
Proposed Model	**93.8%**	**98.0%**	**96.1%**
Previous Work	91.3%	87.6%	93.8%

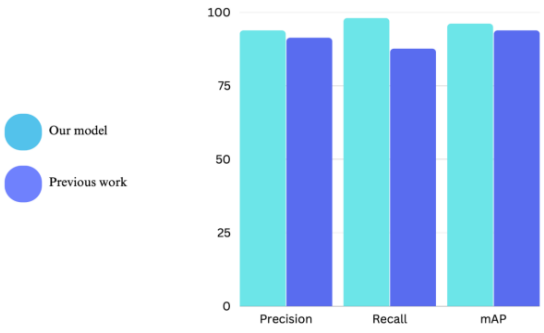

FIGURE 3: Results Comparison

Confusion Matrix:
A confusion matrix is a chart used to assess the effectiveness of a classification model. It's a matrix containing actual values in the rows and predicted values in the columns, with each cell representing the number of times the actual and anticipated values match or don't match. The Figure 4 represented the confusion matrix of proposed model result.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

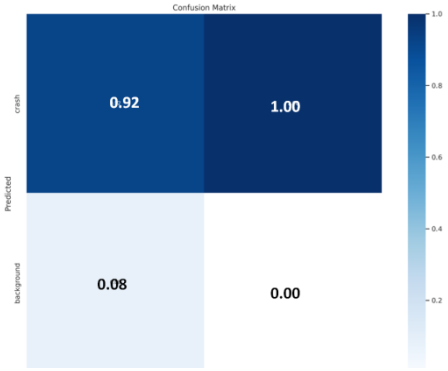

FIGURE 4: Confusion Matrix of Our Model

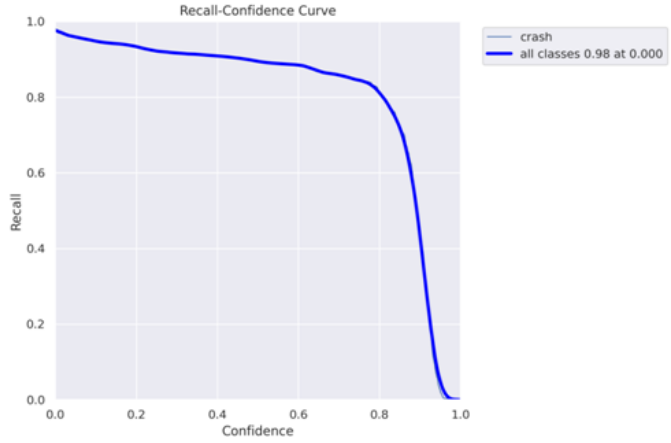

FIGURE 6: Recall of proposed ensemble Model

Precision:

Precision is a machine learning performance indicator that estimates the proportion of true positives (properly recognized positive samples) among all positive samples recognized by the model. It's frequently utilized in binary classification problems with two possible classes: positive and negative. A high precision score suggests that the model correctly identifies positive samples and has a low false positive rate. The following Figure 5 shows the precision graph of pproposed model. The precision score is calculated as follows.

precision = true positives / (true positives + false positives

Authors and Affiliations)

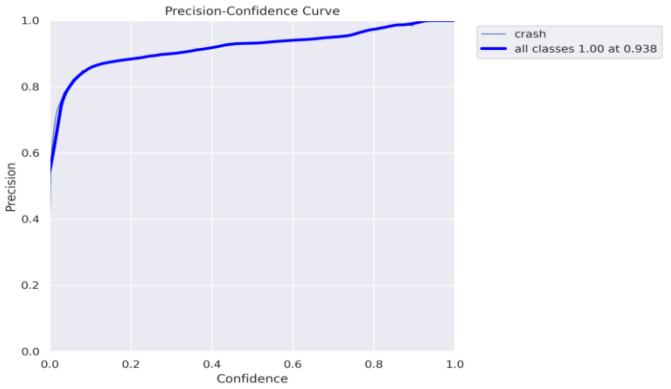

FIGURE 5: Precision Result of proposed ensemble Model

Recall:

Recall is a measure of the completeness or coverage of the model's predictions in machine learning. In particular, recall is defined as the ratio of true positive predictions to the total number of positive cases in the dataset. A high recall score indicates that the model can recognize the majority of positive examples, whereas a low recall score indicates that the model misses many positive cases.

Mathematically, recall can be defined as, recall = true positives / (true positives + false negatives)

mAP:

mAP (mean Average Precision) is a common machine learning performance metric for object detection and image segmentation tasks. The mAP is computed by averaging the recall at various IoU (Intersection over Union) levels. The Figure 7 repreented the mean average precision graph of ensemble model.

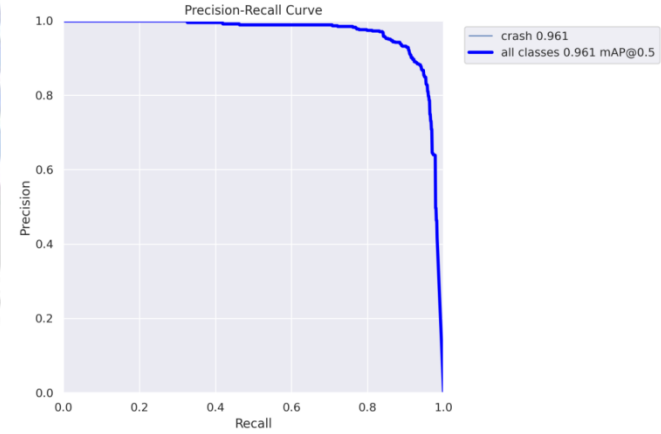

FIGURE 7: mAP of Our Model.

Custom model training:

Training process of ensemble model is described here. The model trained for 50 epochs, with an input image size of 640x640 pixels. Details on the training process, including the total time taken to train the model, as well as the results obtained after every 10 epochs is represented in TABLE 4 and Figure 8. The purpose of this Custom model training is to give a comprehensive overview of the model training process, as well as to provide insights into the effectiveness of the training strategy used

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

TABLE 4: Custom model training

Epoch	box_loss	Precision	Recall	mAP
1 / 50	1.121	0.696	0.69	2
10 / 50	1.242	0.736	0.686	0.729
20 / 50	1.043	0.86	0.828	0.889
30 / 50	0.8808	0.906	0.858	0.919
40 / 50	0.8054	0.921	0.884	0.947
50 / 50	0.5169	0.905	0.917	0.962

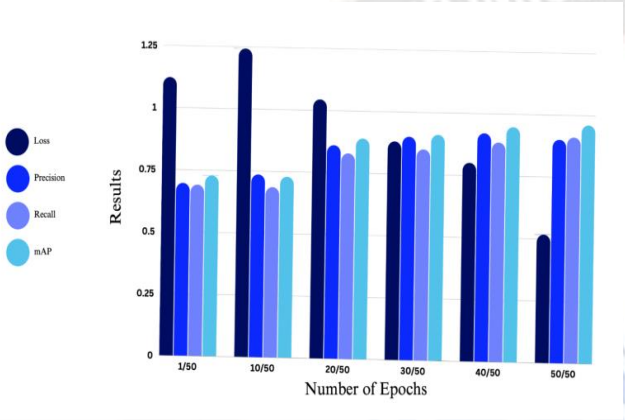

FIGURE 8: Graph showing custom model results.

The log output from a YOLOv8 object detection model, which has been trained for 50 epochs in 1.471 hours, includes some details about the YOLOv8 model. It has 268 layers, 43607379 parameters, 0 gradients, and can perform 164.8 GFLOPs. The model is using Python-3.9.16 and torch-1.13.1+cu116, and is running on a Tesla T4 GPU with 15102MiB of memory. The model has been evaluated on 510 images and has detected 516 instances of objects. The precision and recall for the detected objects are 0.93 and 0.98, respectively, and the mean average precision (mAP) at a threshold of 0.5 is 0.961. The mAP between a threshold of 0.5 and 0.95 is 0.76. The speed of the model is also provided, with 1.0ms for preprocessing, 16.2ms for inference, 0.0ms for loss, and 1.7ms for post processing per image

VI. Conclusion

In this paper, we proposed an ensemble model which offers an approach for real-time car crash detection in traffic surveillance utilizing YOLOv8 and OpenCV. Our experimental results show that the proposed approach is efficient in reliably detecting and localizing car crashes in real-world scenarios with high precision and recall. Results show that the proposed model can identify car crashes with an accuracy of 93.8% and a mean average precision (mAP) of 96.8%. The proposed technique has significant implications for enhancing the efficiency and reliability of traffic surveillance systems, which can help minimize traffic accidents and enhance overall road safety. However, our findings can be used in various scenarios, including highway and crossroads monitoring, automated emergency response systems, and autonomous navigation.

In addition to the obvious practical benefits of our work, our discoveries can pave the way for future research. One key future research area is to investigate the use of additional sensor modes, such as lidar or radar, to supplement the visual data obtained by the camera. These sensors can provide additional information regarding the location and velocity of cars, which can increase crash detection accuracy. Another significant future research direction is applying transfer learning to adjust our model to diverse camera setups and environmental circumstances. Transfer learning allows us to use what we learned from one dataset to improve the performance of another, thereby lowering the quantity of data required for training and increasing the model's generalization. Finally, it would be interesting to look into using generative models for data augmentation and synthetic data production, such as generative adversarial networks (GANs), which helps us overcome the constraints of real-world data and train more robust and accurate models.

To summarize, the proposed ensemble model produced remarkable accuracy in traffic surveillance and highlights the potential for using deep learning and computer vision techniques to improve road safety. The proposed method can easily integrate into existing traffic surveillance systems and has substantial implications for minimizing traffic accidents and saving lives. Our findings will encourage additional research in this field and lead to the development of more advanced and effective traffic surveillance systems.

References

[1] Wang, Y., Huang, Y., Li, X., & Liu, Z. (2020). A Review of Machine Learning Approaches for Traffic Incident Detection and Management. IEEE Access, 8, 202359-202372. doi: 10.1109/ACCESS.2020.3035549.

[2] Suriya, N. C., Immanuel, J., & Balaji, R. (2020). An overview of the YOLO algorithm for traffic accident detection and analysis. International Journal of Advanced Science and Technology, 29(9), 5597-5605. doi: 10.1007/978-3-030-58805-2_22.

[3] Taha, M. I., & Almohaimeed, A. (2014). Real-time traffic accident detection and management system. 2014 IEEE International Conference on Industrial Engineering and Engineering Management, 1191-1195. doi: 10.1109/IEEM.2014.7058805.

[4] Lu, K., Lin, D. D., & Loo, C. K. (2014). Real-time automatic detection of traffic accidents in surveillance video. 2014 IEEE

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5s
DOI: https://doi.org/10.17762/ijritcc.v11i5s.6651
***Article Received**: 28 February 2023 **Revised**: 09 April 2023 **Accepted**: 25 April 2023*

International Conference on Multimedia and Expo Workshops (ICMEW), 1-6.
doi: 10.1109/ICMEW.2014.6890511.

[5] Nogueira, A. L., & Oliveira, M. M. (2014). Automatic detection of traffic accidents from closed-circuit television footage. 2014 IEEE International Conference on Image Processing (ICIP), 3477-3481.
doi: 10.1109/ICIP.2014.7025609.

[6] Mathur, A., Agrawal, R., & Khanna, A. (2015). Real-time vehicle accident detection system using surveillance video analysis. Procedia Computer Science, 70, 641-647.
doi: 10.1016/j.procs.2015.10.076.

[7] M. Rizwan et al. (2016). Real-Time Vehicle Accident Detection System based on Image Processing Techniques. In 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, 2016, pp. 250-255,
doi: 10.1109/FIT.2016.53.

[8] M. Rizwan et al. (2019). Real-time Vehicle Accident Detection System using Machine Learning Techniques. In 2019 IEEE International Conference on Advanced Information Technology, Services, and Systems (AITSS), Marrakesh, Morocco, 2019, pp. 1-6.doi: 10.1109/AITSS.2019.8777166.

[9] Sabrin, S. M., Rahman, M. A., Hassan, M. R., & Hossain, M. S. (2019, September). Real-Time Detection of Road Accidents using Deep Learning Techniques. In 2019 International Conference on Robotics, Electrical, and Signal Processing Techniques (ICREST) (pp. 250-255). IEEE.
doi: 10.1109/ICREST45688.2019.9079712.

[10] Wang, J., Lai, L., & Guo, Y. (2019). A Real-Time Vehicle Detection and Crash Detection Algorithm for Intelligent Transportation Systems. IEEE Access, 7, 24932-24941. doi: 10.1109/access.2019.2907535.

[11] Lee, J., Kim, M., and Kim, C. "Real-Time Traffic Accident Detection using Deep Convolutional Neural Networks." 2019 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2019, pp. 878-883. IEEE. doi: 10.1109/ICARCV.2018.8581111.

[12] Zhang, L., Ren, Y., Li, X., & Wu, Y. (2020). A Real-Time Object Detection Method for Traffic Surveillance System Based on YOLOv3. Applied Sciences, 10(20), 7293.
doi: 10.3390/app10207293.

[13] M. Saleem, et al. (2021) .Traffic Sign Detection and Classification using YOLOv3 and Machine Learning. Processes, vol. 9, no. 4, p. 446, Mar. 2021.
doi: 10.3390/pr9040446.

[14] Yang, Z., Feng, Y., Li, J., & Li, W. (2021). An Intelligent Vehicle Monitoring System Based on YOLOv4 and Cloud Computing. Applied Sciences, 11(11), 5184.
doi: 10.3390/app11115184.

[15] Sun, J., Wang, S., & Wei, P. (2021). Traffic Sign Detection and Recognition using YOLOv3 and Convolutional Neural Networks. Electronics, 10(7), 826.
doi: 10.3390/electronics10070826.

[16] Nguyen, T. D., Nguyen, D. T., and Vo, N. L. (2021). A Real-time Traffic Surveillance System based on YOLOv3 and EfficientNet. 11th International Conference on Communications and Electronics (ICCE), 2021, pp. 423-428. IEEE. doi: 10.1109/ICCE51597.2021.9522188.

[17] The dataset is publicly available at: https://universe.roboflow.com/mada-study/crash-car-detection.

[18] Mane, D.T., Sangve, S.M., Upadhye, G.D., Kandhare, S., Mohole, S., Sonar, S., & Tupare, S. (2022). Detection of Anomaly using Machine Learning: A Comprehensive Survey. International Journal of Emerging Technology and Advanced Engineering. Vol. 12, issue 11, pp. 134-152. DOI: 10.46338/ijetae1122_15