



Article

Visual Detection of Traffic Incident through Automatic Monitoring of Vehicle Activities

Abdul Karim ¹, Muhammad Amir Raza ¹, Yahya Z. Alharthi ², Ghulam Abbas ³, Salwa Othmen ^{4,*},
Md. Shouquat Hossain ⁵, Afroza Nahar ⁶ and Paolo Mercorelli ^{7,*}

- ¹ Department of Electrical Engineering, Mehran University of Engineering and Technology, SZAB Campus Khairpur Mir's, Khairpur 66020, Sindh, Pakistan; karim@muetkhp.edu.pk (A.K.); amirraza@muetkhp.edu.pk (M.A.R.)
- ² Department of Electrical Engineering, College of Engineering, University of Hafr Albatin, Hafr Al Batin 39524, Saudi Arabia; yalharthi@uhb.edu.sa
- ³ School of Electrical Engineering, Southeast University, Nanjing 210096, China; lashariabbas@seu.edu.cn
- ⁴ Department of Computers and Information Technologies, College of Sciences and Arts Turaif, Northern Border University, Arar 91431, Saudi Arabia
- ⁵ Department of Electrical and Electronic Engineering, International University of Business Agriculture and Technology (IUBAT), Dhaka 1230, Bangladesh; engi.shouquat@gmail.com
- ⁶ Department of Computer Science, American International University-Bangladesh, Dhaka 1229, Bangladesh; afroza@aiub.edu
- ⁷ Institute for Production Technology and Systems (IPTS), Leuphana Universität Lüneburg, 21335 Lüneburg, Germany
- * Correspondence: salwa.othmen@nbu.edu.sa (S.O.); paolo.mercorelli@leuphana.de (P.M.)

Abstract: Intelligent transportation systems (ITSs) derive significant advantages from advanced models like YOLOv8, which excel in predicting traffic incidents in dynamic urban environments. Roboflow plays a crucial role in organizing and preparing image data essential for computer vision models. Initially, a dataset of 1000 images is utilized for training, with an additional 500 images reserved for validation purposes. Subsequently, the Deep Simple Online and Real-time Tracking (Deep-SORT) algorithm enhances scene analyses over time, offering continuous monitoring of vehicle behavior. Following this, the YOLOv8 model is deployed to detect specific traffic incidents effectively. By combining YOLOv8 with Deep SORT, urban traffic patterns are accurately detected and analyzed with high precision. The findings demonstrate that YOLOv8 achieves an accuracy of 98.4%, significantly surpassing alternative methodologies. Moreover, the proposed approach exhibits outstanding performance in the recall (97.2%), precision (98.5%), and F1 score (95.7%), underscoring its superior capability in accurate prediction and analyses of traffic incidents with high precision and efficiency.

Keywords: object tracking; object detection; traffic incident; sustainable transportation



Citation: Karim, A.; Raza, M.A.; Alharthi, Y.Z.; Abbas, G.; Othmen, S.; Hossain, M.S.; Nahar, A.; Mercorelli, P. Visual Detection of Traffic Incident through Automatic Monitoring of Vehicle Activities. *World Electr. Veh. J.* **2024**, *15*, 382. <https://doi.org/10.3390/wevj15090382>

Academic Editors: Ayman EL-Refaei, Stergios Mavromatis, Yasser Hassan and George Yanniss

Received: 12 July 2024

Revised: 8 August 2024

Accepted: 21 August 2024

Published: 23 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rise in car ownership and road traffic due to modernization has led to several transportation and traffic control issues like severe traffic jams having become more common as the number of vehicles on the roads has increased dramatically [1]. This congestion wastes time, fuel, and money for drivers. The higher volume of traffic has contributed to a rise in car accidents and crashes [2]. More vehicles on the roads, especially in urban areas, means a greater risk of collisions occurring [3]. Increased car usage has negative environmental impacts, including higher air pollution and greenhouse gas emissions from vehicle exhaust. This worsens air quality and contributes to climate change [4]. The need for more parking spaces has led to the construction of large parking lots and garages, which can take up valuable urban land that could be used for other purposes [5]. Traffic noise pollution has risen as a result of the higher number of vehicles, negatively impacting quality

of life for those living near busy roads. Maintaining and expanding road infrastructure to handle the increased traffic requires significant public spending on construction and repairs [6].

A number of problems appear as a result of road traffic accidents globally: approximately 1.35 million fatalities each year and millions more seriously injured or disabled [7]. Road traffic injuries are the leading cause of death for children and young adults aged 5–29 years [8]. Over 90% of road traffic deaths occur in low- and middle-income countries, even though these countries have only about 60% of the world's vehicles [9]. Road traffic crashes cost most countries around 3% of their gross domestic product in medical costs, lost productivity, and other expenses [10]. Accidents have significant economic impacts, including healthcare costs, loss of productivity, increased insurance and legal costs, property damage, and strains on emergency services [11]. Accidents cause profound emotional trauma for families and communities, as well as psychological issues like PTSD, anxiety, and depression for survivors. Injuries can lead to temporary or permanent disabilities that reduce quality of life and financial stability for victims and their families [12]. Over 50% of road traffic deaths are among vulnerable road users like pedestrians, cyclists, and motorcyclists [13].

The development of autonomous vehicles and smart cities is closely linked to road safety. Road safety can be improved using autonomous vehicles, which ultimately reduces the accidents [14]. It is found that human error caused 94% of road accidents; however, autonomous vehicles have the capability to avoid accidents at around 1/3 [15]. Advanced technologies based on sensors, artificial intelligence, and surroundings connectivity would help to design autonomous vehicles, which leads to a more secure environment [16]. One example of advanced technology is machine learning, which helps to analyze the received signal from the sensors and make some real and timely decisions for reducing accidents [17]. By using the sensor network, the road condition data and real-time traffic environment should be captured in the autonomous vehicles for adjusting the speed of the vehicle and monitor the roads accordingly where all types of vehicles are moving with different speeds [18,19]. Alongside this, AI helps to react quickly before any danger and resolve the decision making problems for the driver. Furthermore, traffic flow can be managed through communication among autonomous vehicles. Autonomous vehicles can be equipped with adaptive cruise control and lane keeping, to further enhance safety [20].

Moving and immovable objects can be easily detected from the driving ends [21]. Moving objects are human beings (passengers/drivers), wild and domestic animals, busses, cars, bicycles, motorcycles, and many other objects that are moving on the roads. On the other hand, some immovable objects are stationary devices that are used to regulate the traffic flow and guide the driver on when they should slow down, stop, or go with the uniform speed [22]. These two objects are very necessary for smart cities and autonomous vehicles to classify and detect the exact object location in the environment and facilitate the efficient and safe movement of vehicles.

Conventional technologies like Viola–Jones Detector (VJD) of 2001, Histogram of Oriented Gradients (HOG) of 2006, and Deformable Part Models (DPMs) of 2008 are considered to be computer vision technology [23]. VJD is used to detect the objects using a sliding window approach and Haar wavelet features, HOG is used to detect the objects using a feature descriptor, and DPM detects the pedestrians using images [24]. However, advanced technologies like deep learning- and machine learning (YOLO)-based ones are used currently, which accurately detect and classify objects in real time, enabling proactive traffic management [25]. A comparative analysis of advanced technologies for object detection is given in Table 1. In the previous literature, up to YOLOv7 is used for traffic incidents but this study used the YOLOv8 model to configure this study. However, the latest version of YOLOv9 and YOLOv10 has also been developed but their applications have not matured yet to take on traffic incident detection issues. Computer vision-based algorithms can identify and monitor vehicles on the road, providing data on traffic density and flow [26]. Computer vision can recognize traffic signs and lights, enabling automated enforcement of

traffic rules and regulations. Computer vision systems can identify pedestrians, cyclists, and other obstacles, alerting drivers and traffic managers to potential hazards. Computer vision-based algorithms can rapidly detect accidents, stalled vehicles, and other traffic incidents, allowing for faster emergency response. Computer vision can identify unusual traffic patterns or behaviors that may indicate an impending incident, enabling proactive intervention [27]. Hence, the method adopted in this study follows a sequence of steps as listed below:

1. Video streams from road surveillance cameras are collected as input data using Roboflow.
2. Video frames are preprocessed to enhance image quality and reduce noise through Deep-SORT.
3. The YOLOv8 model is initialized with pre-trained weights obtained from a large dataset. The model is fine-tuned using the labeled dataset. After training, the model is capable of real-time accident detection.
4. Upon accident detection, immediate alerts are generated and directed to relevant authorities or integrated into broader traffic management systems.

Table 1. Comparative analysis of advanced technologies (YOLO) for object detection.

S. No.	Reference	Study Year	Study Model	Study Purpose
1	[28]	2023	YOLOv2	This article proposes an approach for detecting Chinese traffic signs using a deep convolutional network.
2	[29]	2021	YOLOv2	In order to address the traditional traffic incident detection, YOLOv2 algorithm is proposed in this study.
3	[30]	2021	YOLOv2	In this research, an enhanced model of YOLOv2, which aims to address the shortcomings in its inability to recognize small targets, is proposed. The enhanced model is able to identify more little things than the original model for the same image that contains small objects. This new approach could identify items more reliably in photos with complicated backgrounds. To put it briefly, this enhanced model becomes more sensitive to small objects and improves recognition accuracy.
4	[31]	2020	YOLOv3	Vehicle detection using images and video capturing is an important task for sustainable transportation. However, to achieve this, YOLOv3-DL model is built on the Tensorflow framework.
5	[32]	2020	YOLOv3	Traffic sign detection scheme is proposed in this study using YOLOv3 for real-time detection with high precision.
6	[33]	2021	YOLOv4	With enough annotated training data, Convolutional Neural Networks (CNNs) reach the pinnacle of traffic sign identification. The dataset uses CNN to assess the overall visual system's quality. Sadly, there are not many databases available for traffic signs from most countries in the world. In this case, more realistic and diverse training images could be generated via Generative Adversarial Networks (GANs) to complement the real image arrangement.
7	[34]	2022	YOLOv4	This research analyzes object detection techniques like Yolo V4 and Yolo V4-tiny merged with Spatial Pyramid Pooling (SPP). In this work, the significance of the SPP principle is assessed in terms of improving the efficiency with which Yolo V4 and Yolo V4-tiny backbone networks extract features and learn object features.
8	[35]	2021	YOLOv4	YOLOv4 model is proposed in this study for making accurate detection of traffic incidents to avoid accidents.

Table 1. Cont.

S. No.	Reference	Study Year	Study Model	Study Purpose
9	[36]	2021	YOLOv5	Digital driving system is proposed using YOLOv5 model that predicts the multi-scale objects in the traffic.
10	[37]	2022	YOLOv5	A lot of conjecture has recently surrounded advanced driver-assistance systems (ADASs), which give drivers the greatest possible driving experience. Today's traffic accidents are often caused by unsafe driving conditions, which are detected by ADAS technology.
11	[38]	2021	YOLOv5	The ability to identify irregularities like traffic accidents in real time is proposed in this study for intelligent traffic monitoring system using deep learning approach.
12	[39]	2022	YOLOv6	There are a lot of accidents and long lines of traffic on Indian roads these days. All things considered, traffic management is a crucial issue that affects us frequently. Utilizing expertise, such as IoT and image processing, can facilitate the movement of an efficient traffic monitoring system. In order to prevent collisions between cars during traffic signals, we can assess the density of the traffic and plan the flow of vehicles at crosswalks such that no collisions occur and traffic on both sides of the road is given equal priority.
13	[39]	2022	YOLOv6	Pothole detection tests have demonstrated the immense potential of CNNs using YOLOv6 as the main objective of this study.
14	[40]	2023	YOLOv7	Considering when cars, pedestrians, and micromobility vehicles collide at right angles on an urban road network, the authors took pedestrian crosswalks into consideration. These road segments are places where automobiles pass perpendicular to the path of vulnerable individuals. It is intended to provide a warning system for cars and pedestrians in these locations to prevent accidents. This process involves several steps, including concurrently alerting drivers, people with disabilities, and distracted pedestrians to the dangers of cell phone addiction.
15	[41]	2023	YOLOv7	In computer vision, traffic sign detection is an essential job with broad applications in autonomous driving. This work provides a small-object detection technique for traffic signs based on the modified YOLOv7.
16	[42]	2024	YOLOv7	A possible substitute for pothole detection could be a deep learning- and computer vision-based method. In order to identify different roadblocks, the suggested system uses the CNN with YOLOv7 algorithms.
17	Our Proposed Study	2024	YOLOv8	None of the earlier studies utilized YOLOv8 model for traffic incident detection, which can handle the challenging task given the dynamic nature of urban traffic and the multitude of events that can occur. However, in this study, Roboflow is used for the data compilation and preparing the image data for computer vision models. The initial dataset comprised 523 images, with 335 images designated for training, 144 for validation, and 44 for testing purposes. Then, Deep Simple Online and Real-time Tracking (Deep-SORT) algorithm is developed to scrutinize scenes at different temporal layers and provide continuous information about vehicular behavior. Then, YOLOv8 model detects the actual traffic incident.

2. Research Method

In this study, a comprehensive system designed to enhance driving safety and efficiency through the integration of various components is presented. The suggested approach combines real-time data collection, predictive modeling, and a rule-based prediction system. This paper presents a novel approach to object detection, focusing on identifying vehicles such as cars, trucks, and buses within specific regions. Modified YOLOv8 is utilized for object detection and through traffic cameras. Figure 1 shows the conceptual block diagram of the proposed methodology. RoboFlow was used to collect the data, Deep-SORT scrutinized the collected data, and then YOLOv8 was implemented for an accurate detection of a traffic incident. Upon a successful detection of a traffic incident, an immediate alert is generated and directed to relevant authorities.

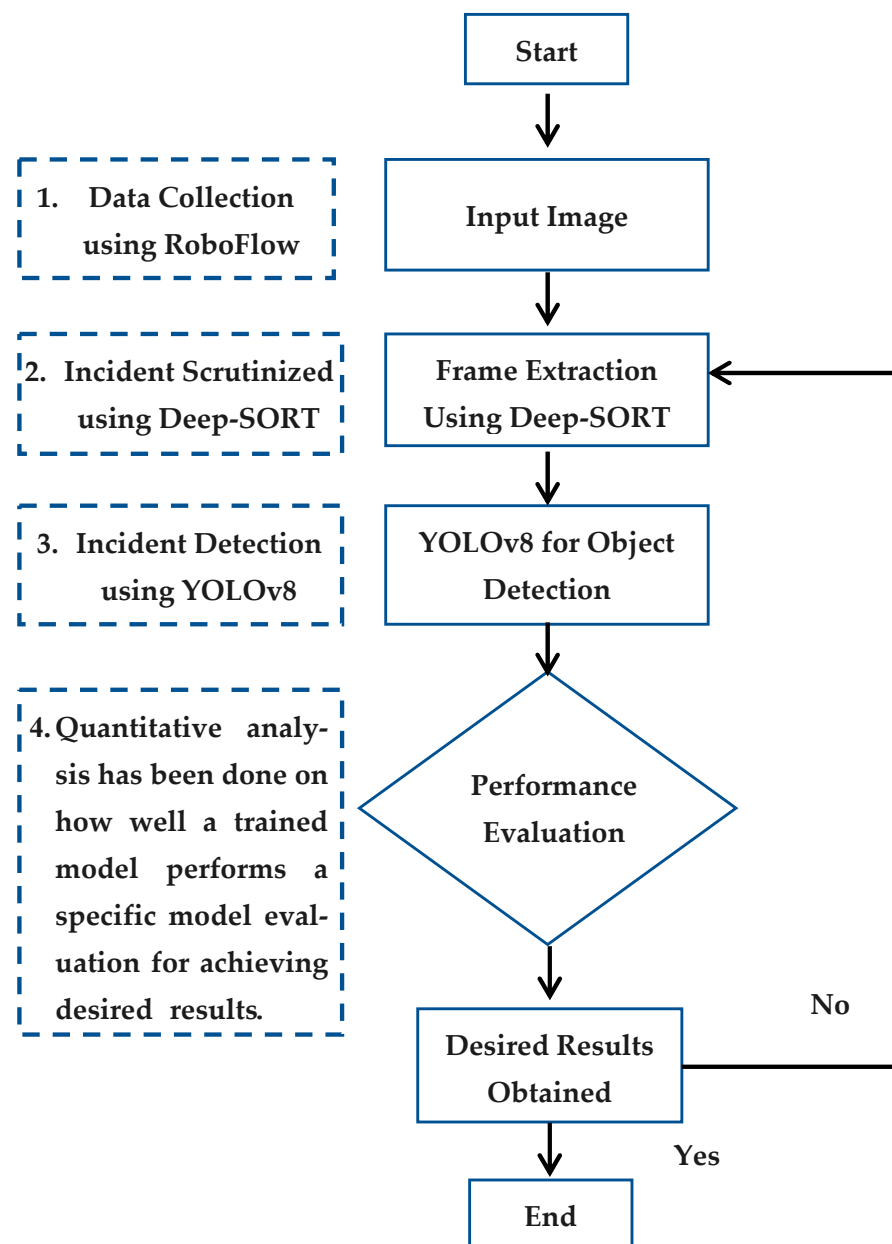


Figure 1. Conceptual block diagram of proposed method.

RoboFlow has the capability to manage the dataset of accident detection and provides accurate and real-time information about accidents on the roads [43]. Deep SORT builds upon the SORT algorithm, which excels in tracking precision and accuracy but struggles

with identity switches and occlusions [44]. Deep SORT addresses these limitations by introducing a better association metric that combines motion and appearance descriptors [45]. Before tracking objects, Deep SORT trains a feature embedding model on a large-scale dataset to create a well-discriminating feature space [46]. Cosine metric learning is used in Deep SORT to train the model. Deep SORT uses spatial and temporal features in videos for locating the exact object tracking and it maintains the object IDs [47]. Furthermore, the integration of Deep-SORT with YOLOv8 is possible because YOLOv8 is used to detect the object and Deep-SORT handles tracking.

The deep learning-based YOLOv8 model is used to detect the object and its applications are diverse including road accident detection [48]. In road accidents, YOLOv8 has the upper hand among all other models and has achieved optimal performance [49]. The CSPDarknet53 architecture in YOLOv8 is considered to be the backbone network for feature extraction, balancing accuracy and speed [50]. Three detection heads of varying scales are employed to detect objects of different sizes effectively. The model is trained on a labeled dataset containing images and annotations of road accidents. The training process involves optimizing the model's parameters to minimize detection errors and improve accuracy [51]. Parameters such as the batch size, learning rate, anchor boxes, IoU threshold, confidence threshold, and NMS threshold are tuned to achieve optimal performance [52]. As the name implies, YOLO estimates the bounding boxes and class probabilities from the image pixels with just one glance. This unified detection framework is perfect for real-time object detection systems since it is incredibly fast and end-to-end-differentiable [53]. Figure 2 illustrates how each bounding box was confined to recognizing a single object, hence limiting the first iteration of YOLO [54].

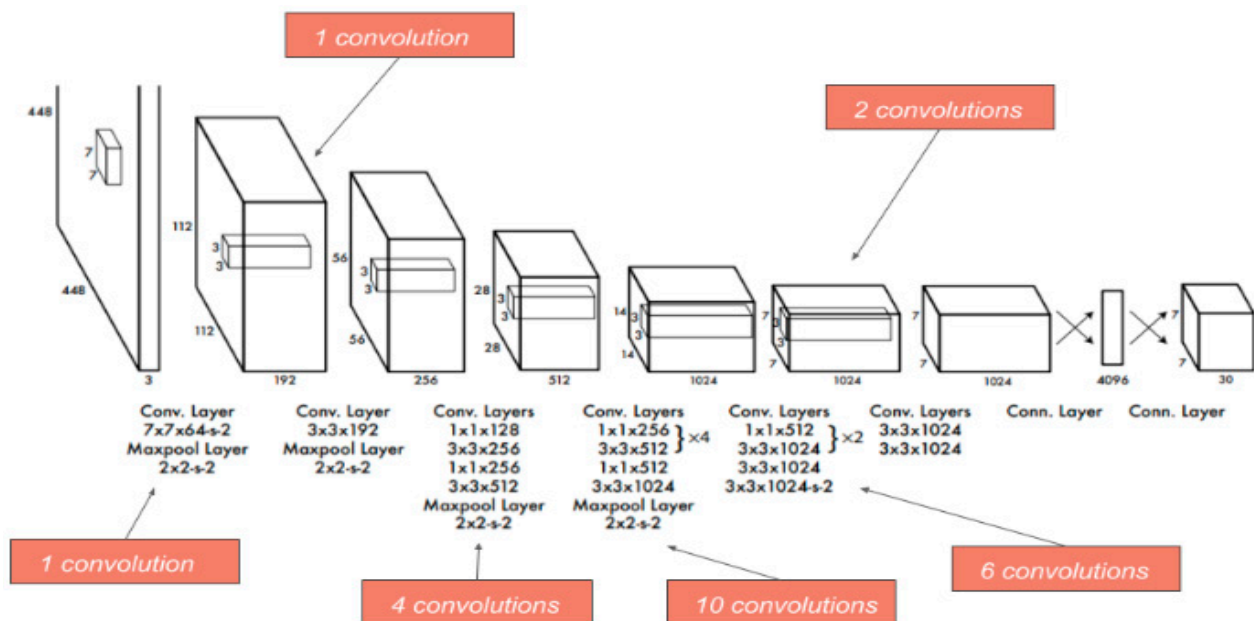


Figure 2. YOLO basic architecture [54].

YOLOv2, or YOLO9000, was an enhancement of YOLO in a number of areas. The addition of multi-scale training has improved the model's ability to estimate bounding boxes. Additionally, a novel 19-layer architecture called Darknet-19 was presented, which serves as the foundation for feature extraction [55]. YOLOv2 integrated the advantages of the Faster R-CNN and YOLO frameworks, enabling the detection of over 9000 distinct object kinds [56]. By including three additional scales at which detections were produced, YOLOv3 improved upon YOLOv2 [57]. This is similar to the idea of feature pyramid networks. The capacity to detect at various scales made smaller objects simpler to identify. Because YOLOv3 employed three distinct anchor box sizes for every scale, bounding box

prediction was enhanced [58]. It also made use of a recently created network architecture called Darknet-53, a blend of ResNet and Darknet-19 [59].

YOLOv4 included several novel techniques to increase object identification speed and accuracy [60]. The SAM block, PANet, and CSPDarknet53 backbone architecture were among its characteristics. It also used the Mish activation function and the CIOU loss for better performance [61]. One of the best real-time object identification methods is YOLOv4, which greatly improved detection accuracy and speed. The community produced YOLOv5, a decent version of the YOLO series, in an effort to enhance YOLO's accuracy, portability, and speed [62]. The basic block diagram of YOLOv5 is given in Figure 3 [63].

Overview of YOLOv5

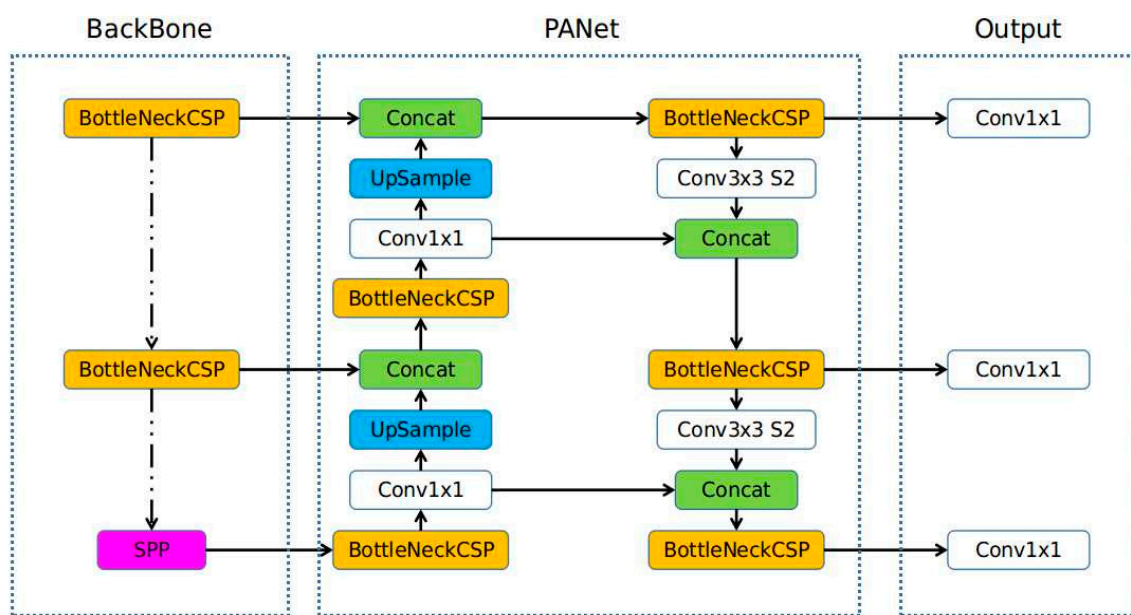


Figure 3. YOLOv5 block diagram [63].

This backbone is used to extract features from the input photos. It is composed of convolutional layers that generate several feature maps by processing the input through a variety of filters. These feature maps capture significant information about the image, including its boundaries, textures, and forms [64]. YOLOv8's real-time object detection is made possible by the backbone's efficient and timely architecture. It can also be made to receive input in many sizes and scales due to its versatility. As a result, YOLOv8 can be used in many different contexts, including small-scale object recognition in images and large-scale video monitoring. The detailed block diagram of YOLOv8 is given in Figure 4 [65].

Predictions based on the traits that the backbone extracts have are made by the YOLOv8 head. It is composed of bottleneck, SPF, convolutional, and c2f blocks, among other layers. The head's convolutional layers refine the features even more and get them ready for the last prediction step by applying more filters to the feature maps [66,67]. The 2D feature maps are converted into a 1D vector by the c2f (convolution to fully connected) layers so that the fully connected layers can analyze it [68]. By lowering the dimensionality of the features, the bottleneck layers serve to increase the model's efficiency and avoid overfitting [69]. Predicting the scale of the items in the image is the job of the SPF (scale prediction feature) blocks, and it is essential for precise object detection. YOLOv8 is superior to YOLOv54 in a number of ways, including the following: The C2f module was used in lieu of the C3 module. The purpose of this upgrade was to increase the model's efficiency. Furthermore, 3×3 convolution is preferred as compared with the 6×6 convolution [70]. The descriptive performance analysis of YOLO models is given in Table 2.

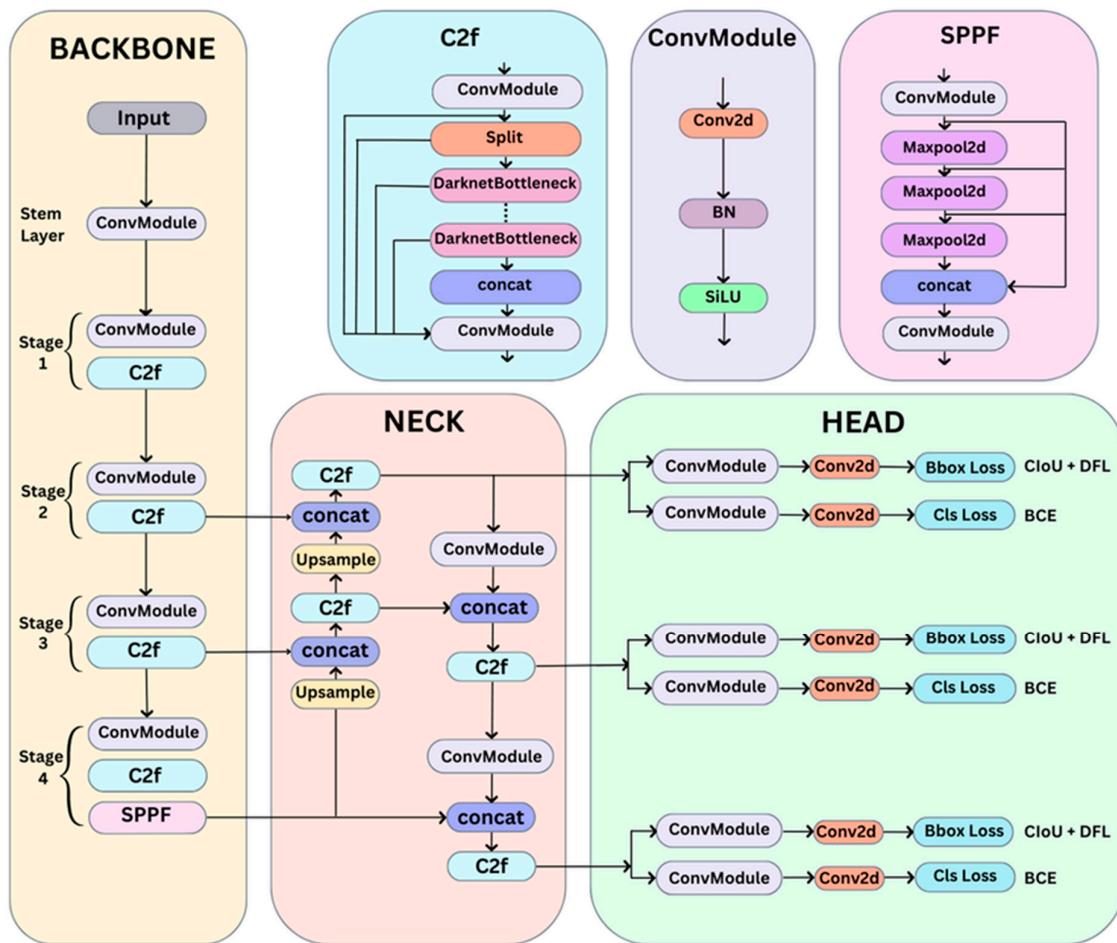


Figure 4. Detailed block diagram of YOLOv8.

Table 2. Comparison between Yolo-based models.

Model	Accuracy (mAP)	Speed (FPS)
YOLOv4	High	Moderate
YOLOv5	High	High
YOLOv6	High	High
YOLOv7	Very High	Moderate
YOLOv8	Very High	Very High

2.1. Setting for Yolo Implementation

The YOLOv8 model can be implemented easily as follows: The dataset should be collected using images and videos; then, preprocess the data by resizing, normalizing, and augmenting the images to enhance the quality and reduce noise. Then, by using the backbone network, configure the YOLOv8 model. For achieving good performance, the following parameters should be monitored accurately, the batch size, anchor boxes, detection heads, learning rate, and confidence threshold, and should be implemented. The preprocess dataset should be used to train the YOLOv8 model through OpenCV and Google Colab and fine-tune the model for reducing the errors and enhancing accuracy. Integrate the trained YOLOv8 model into a system that can analyze video streams from surveillance cameras. Real-time analyses on each frame, detecting accidents, and generating alerts should be performed using the YOLOv8 model. Further, implement the system on highways and monitor the sustainable operation of transportation.

2.2. Dataset

The authors used a publicly accessible dataset of 523 traffic incident photos from the Kaggle public library for this study. A visual depiction of a portion of these photographs that are used in the dataset is shown in Figure 5. The authors used bounding boxes to precisely designate the traffic event in the Pascal visual object class, which allowed them to localize the incident sites within photos. Each image's annotations were kept in separate extensible markup language (XML) files. On the other hand, all classes in the dataset were described in a different protobuf text file. Yet, the YOLOV8 model requires annotations in the COCO dataset format, which is structured as a JavaScript Object Notation (JSON) file. The data were then split up by the authors into training, validation, and testing sets at a ratio of 70:20:10. After that, the authors transformed the Pascal VOC annotations into the COCO format, combining all of the image annotation data into a single JSON file. The authors trained the YOLOV8 model using the preprocessed and divided dataset.

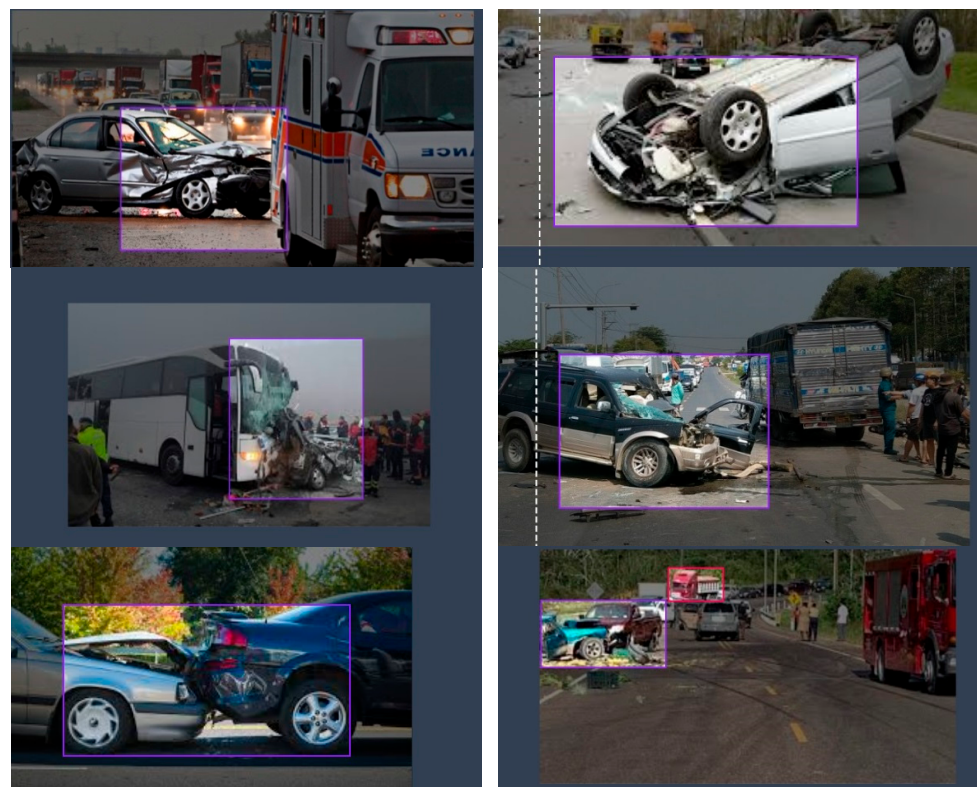


Figure 5. Dataset and data annotation.

2.3. Data Annotation

Data annotation is a common way to tag or label data to make it easier for models to understand. Metadata must be added to raw data in order to boost its value and the computer systems' ability to access it. In computer vision, data annotation is the process of labeling or tagging images to show which features or objects are present or not. In this instance, as Figure 5 illustrates, the incidents were tagged to inform the YOLO model about the traffic event. In order to provide computer vision models with the information they need to learn and recognize objects, data annotation is essential. This study uses RoboFlow that offers a quick and easy method for labeling images for computer vision. With its easy-to-use interface, users may quickly annotate photos with bounding boxes, polygons, and points, among other annotation kinds. Using RoboFlow, data annotation was carried out by uploading the image in the dataset. The Pascal VOC annotation type was then chosen. The annotated photos were exported in COCO format once annotation was finished because YOLO needs this format. The pothole detection algorithm was then trained and tested using these annotated photos.

2.4. Data Augmentation

Data augmentation was used to create more distinctions of current data with diverse features. This technique drops the chance of overfitting while enhancing the model's ability to generate generalizations. This improves the model's accuracy and dependability when forecasting data from the real world. There were 523 photos in the original dataset; 335 were used for training, 144 for validation, and 44 for testing. To improve the dataset's diversity and enrichment, data augmentation techniques like zooming, contrast correction, and image rotation were used. Consequently, 756 photos were added to the dataset. The enhanced dataset was divided using a 70:20:10 split into three comparable sets for testing, validation, and training.

2.5. Model Training

Forward propagation is used to depict the results. An image is sent into forward propagation, which then uses its layers to process the input and provide an output that includes the locations and classes of the objects in the image. By comparing the prediction to the actual value, the loss function is used to determine the model's accuracy. Once the loss has been calculated, the model could start to learn from its mistakes. This kind of propagation is known as reverse/back propagation. During backward propagation, the model calculates the gradient of the loss with respect to each of its parameters. Given that the gradient points in the direction of the loss's steepest climb, the model can lower the loss by changing the parameters in the other direction. The amount that the parameters are altered at each training step is determined by the learning rate, a hyper-parameter that controls the pace of learning. High learning rates have the potential to cause non-convergence and overshooting of the optimal parameters in addition to speeding up the model's learning process.

The model may learn more slowly at a low learning rate, but it will also converge more consistently. Repetition and Periods: For every picture in the training dataset, the steps of forward propagation, loss estimation, and backward propagation are repeated. One training epoch ends when the model has processed every image in the dataset. Several epochs are usually conducted during training, with the model's parameters being modified following each epoch. Another critical hyper-parameter to select is the number of epochs. Underfitting occurs when there are insufficient epochs in the data, causing the model to be unable to recognize the underlying patterns. Overfitting, in which the model performs badly on fresh data due to its excessive specialization to the training set, can be caused by an excessive number of epochs.

2.6. System Integration

Deep-SORT and the YOLOv8 model are integrated since YOLOv8 is utilized for object identification, estimating bounding boxes and class probabilities for each grid cell and partitioning the input image into grid cells. Thereby, Deep SORT employs the Kalman filter and Hungarian algorithm to preserve the identities of the identified objects and forecast their future locations, thereby merging the two methods to provide precise traffic incident monitoring.

3. Results and Discussion

The model's performance was evaluated using a variety of metrics, including mean average precision (mAP), accuracy, recall, classification loss, and differential loss. The suggested model's accuracy is displayed in Figures 6 and 7 with data labeling and annotation. The model shows the accuracy (98.4%), recall (97.2%), precision (98.5%), and F1 score (95.7%) of the proposed approach, which are greater than traditional methods. The accuracy of the YOLOv8 (98.4%) is greater than Faster R-CNN (87.97%), FA-YOLO (92.95%), and FESSD (83%).

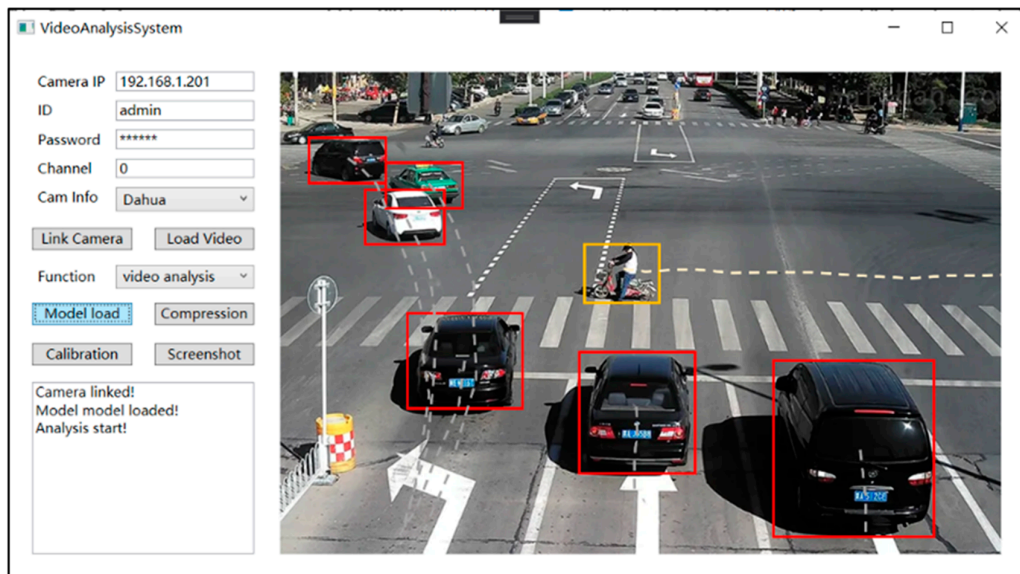


Figure 6. Data labeling and annotation in digital library.



Figure 7. Real-time detection.

3.1. Training and Validation Phase

Throughout the training period, the model displayed a consistent learning trend. The box loss, classification loss, and differential loss all showed a steady drop over 40 epochs. This implies that the model was effectively learning from the training data and improving its ability to recognize automobiles in the images, as seen in Figure 8. In the validation stage, the model kept yielding favorable outcomes. As for the training period, there was a decrease in both the classification loss and the box loss. But up until around epoch 20, the differential loss remained comparatively constant; after that, it began to drastically fall. This means that, contrary to what Figure 8 indicates, the model was not overfitting the training set and was successfully able to generalize to unknown data.

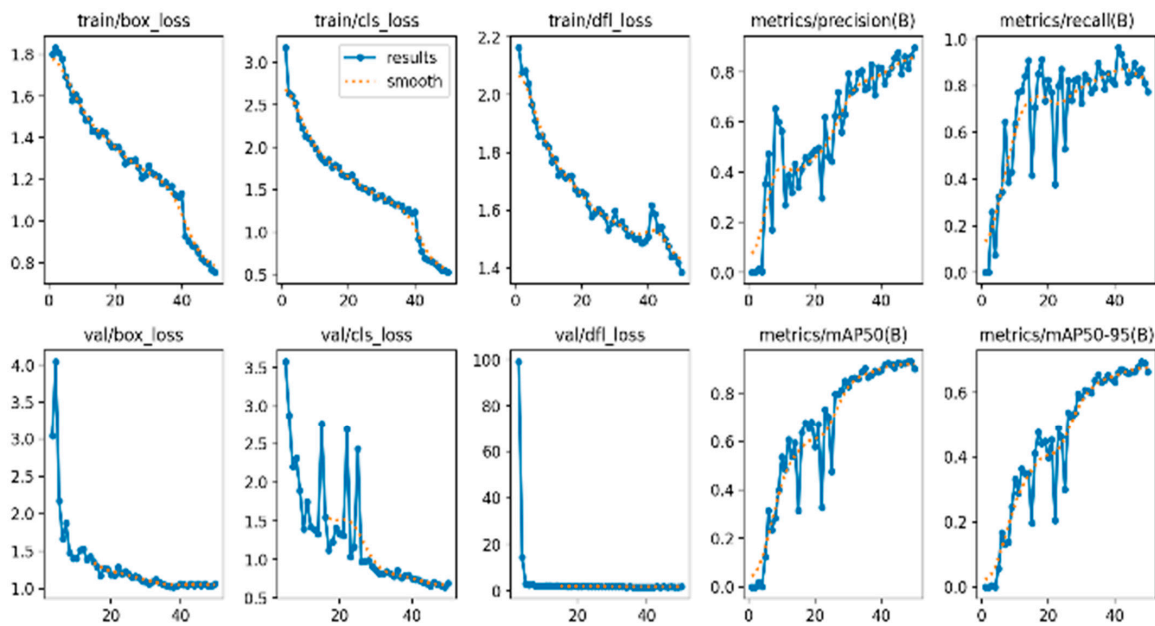


Figure 8. YOLOv8 training losses.

3.2. Recall–Confidence

We evaluated the recall and accuracy metrics as well. These measures showed an overall rising tendency, even if they varied over the epochs. This implies that the model improved in precision (high accuracy) and decreased in the likelihood of missing real cars (high recall) with further training. 4.4 Mean average precision (mAP) and the mAP for IoU = 0.50 showed volatility before stabilizing at about epoch 20, in contrast to the mAP for IoU between 0.50 and 0.95, which, as seen in Figure 9, displays an overall increasing trend. This suggests that the model was improving its bounding box prediction accuracy even for varying degrees of overlap between the predicted and actual bounding boxes.

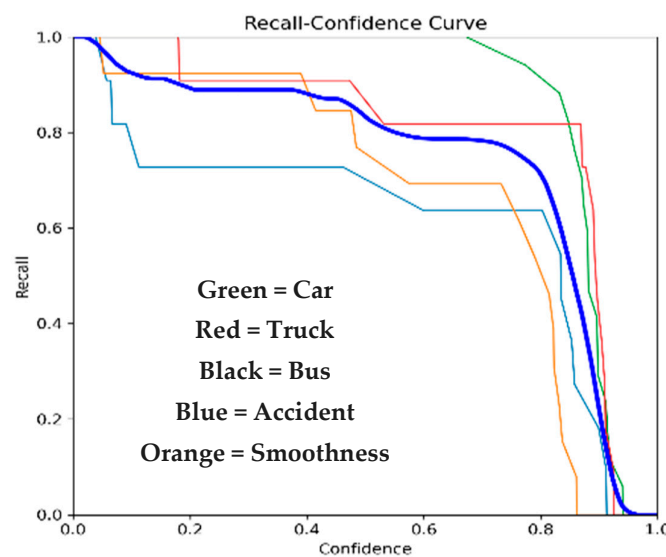


Figure 9. Recall–Confidence curve.

The Recall–Confidence curve is a graphical representation of the model’s performance over a range of confidence levels. The x-axis displays the confidence level, or the probability that the projected label is right. Plotted on the y-axis is recall, or the proportion of true positive events that were correctly identified. Each colored line on the graph represents a different class. The blue line represents the “accident” class, the green line the “car” class,

the red line the “truck” class, and the black line the “bus” class. The jagged form of the lines indicates that changes in confidence levels lead to changes in memory. Throughout the training period, the model displayed a consistent learning trend.

The loss metrics showed that the model was improving its ability to recognize the classes in the photographs and learning from the training data, as they dropped over the course of the epochs. During the validation phase, the model continued to produce positive results. The model was able to generalize effectively to new data and was not overfitting the training set, as evidenced by the loss measures declining in a way similar to that of the training phase. In general, the accuracy and recall metrics showed an increasing tendency, but they varied throughout the epochs. This implies that the model improved in precision (high accuracy) and decreased in the likelihood of missing true positive examples (high recall) as training progressed. The mean average precision (mAP) for IoU = 0.50 varied before settling after epoch 20, as seen in Figure 9. The mAP for IoU, on the other hand, showed an overall growing trend between 0.50 and 0.95. This suggests that the model was improving its bounding box prediction accuracy even for varying degrees of overlap between the predicted and actual bounding boxes.

3.3. Precision–Confidence

Figure 10 shows Precision–Confidence Curve, a graphical representation of the performance of a classification model. Considering the labels “accident”, “car”, “truck”, and “bus”, it seems that this graph can be used for tasks involving the recognition of accidents and object detection. The domains of computer vision and machine learning greatly benefit from this graph. The graph’s x-axis displays the confidence level, or the probability that the predicted label will materialize. Plotted on the y-axis, precision represents the proportion of affirmative identifications that were, in fact, correct. Each of the colorful lines represents a different class. The blue line represents the ‘accident’ class. It shows how different levels of confidence have an impact on how accurate “accident” projections are. Similar to this, the categories “car”, “truck”, and “bus” are indicated by the corresponding red, green, and black lines. The average performance for all courses is shown by the multicolored line that reads “all classes 1.00 at 0.98”. It implies that the model has an accuracy of 1.00 at a confidence level of 0.98 after accounting for all classes.

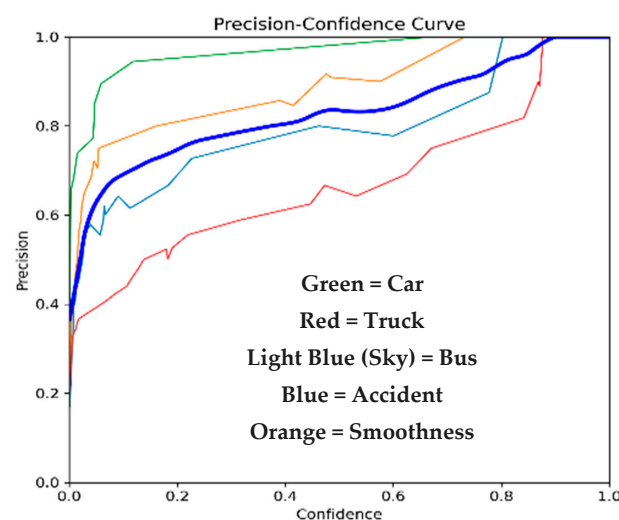


Figure 10. Precision–Confidence Curve.

3.4. Precision–Recall

Recall is the percentage of true positive cases that were correctly detected, and it is shown on the graph’s x-axis. Precision is shown on the y-axis as the percentage of affirmative identifications that were in fact right. Every colored line signifies a distinct class. The blue line represents the ‘accident’ class. It illustrates how different recall capac-

ities impact how accurate “accident” forecasts are. The colorful line labeled “all classes 0.933 mAP@0.5” seems to show an overall performance with the intersection over union (IoU) being 0.5 and the mean average precision (mAP) being 0.933. The precision–recall curve is shown in Figure 11.

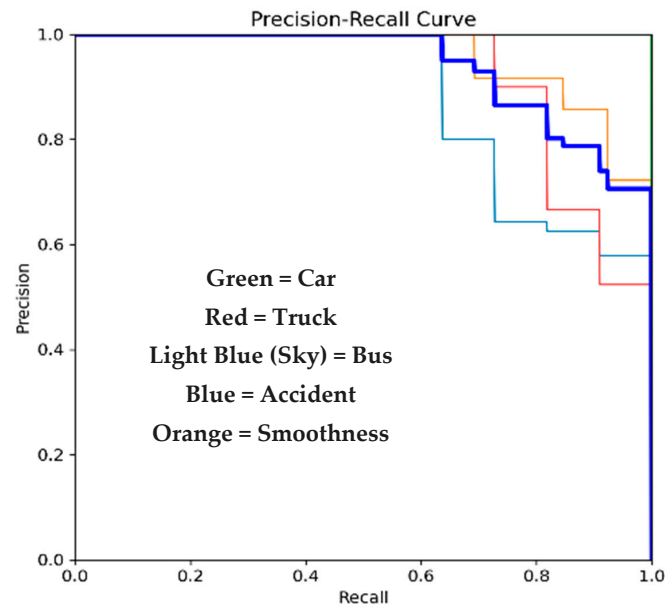


Figure 11. Precision–recall curve.

3.5. F1 Score

The F1 score is a critical metric in evaluating the performance of object detection models like YOLOv8, providing insights into the balance between precision and recall, essential for accurate model assessment. The YOLOv8 model achieved an F1 score of 0.89 or 89% as shown in Figure 12. This F1 score graph helps the fields of computer vision and machine learning immensely. The matrix may be used to calculate a variety of metrics, such as the precision (the number of relevant items selected), recall (the number of relevant objects picked), and F1 score (the harmonic mean of precision and recall).

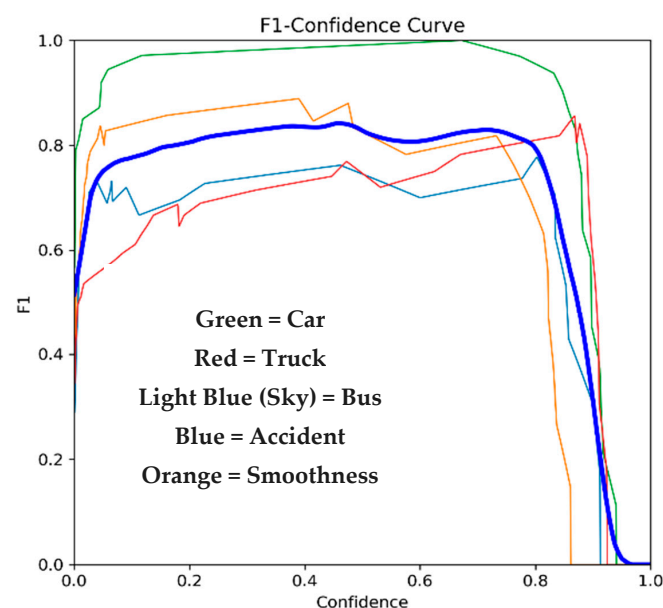


Figure 12. F1–confidence curve.

3.6. Comparative Analysis

Figure 13 displays, on the right-hand side, and offers a comparison of the processing speed of models. This speed is measured by the latency for the given examples. While it may not be immediately clear to discern the specific models from the image, it is generally understood within the field that the latest iterations of YOLO, such as YOLOv8, offer faster and better performance. Therefore, when one considers both the size of the model and the speed of production, it becomes apparent that YOLOv8 strikes a highly complex yet efficient balance. This balance renders it an extremely suitable choice for applications that necessitate real-time object detection and tracking. An example of such an application is the detection of obstacles in urban environments.

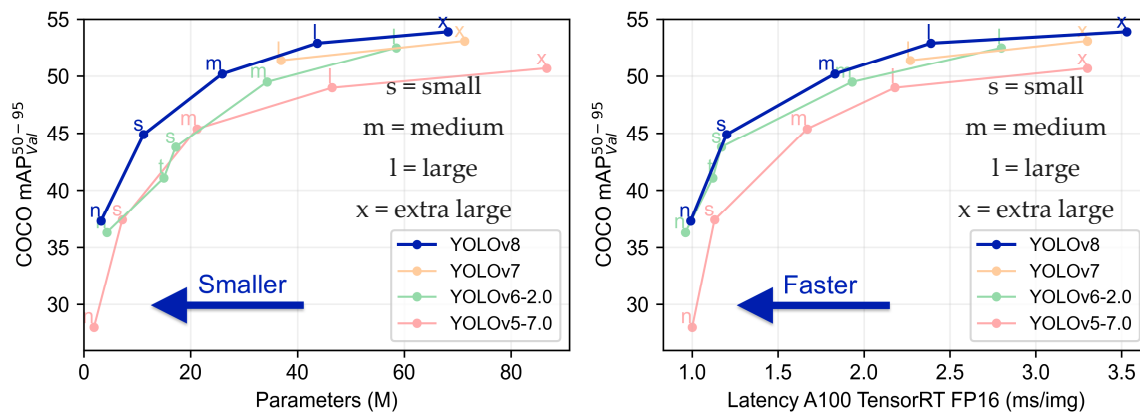


Figure 13. Comparative analysis of YOLO models for processing speed.

Our study covers the latest method for traffic incident detection. However, some of the past literature covers the traditional methods, which are discussed here: Lately, there has been a significant increase in the use of Internet of Things (IoT) systems to shorten the time needed for disaster rescue. This paper proposes an Internet of Things (IoT)-based automobile accident detection and categorization (ADC) system that reports the type of accident in addition to detecting it by combining the built-in and connected sensors of smartphones [71]. Another study [72] offers a process for creating a dependable, computationally cheap, real-time automatic accident detection system that requires the least amount of hardware to be installed. To be more precise, we divided our automatic accident detection system into three main phases: detection, tracking, and classification. We then suggested less computationally demanding solutions for each phase. Paper [73] investigates the viability of applying deep learning models to the detection and prediction of crash risk. For the investigation, data on volume, speed, and sensor occupancy were gathered from roadside radar sensors along Interstate 235 in Des Moines, Iowa. In this study [74,75], we use the Quadrant Scan, a recurrence-based technique, to analyze time series traffic volume data in order to find incidents. Numerous sensors along a stretch of a metropolitan highway record the data.

Furthermore, our study proposed the latest model of YOLO while the latest studies used some older models; the details of some of the latest studies are given: This study [76] uses the YOLOv6 object detection algorithm to identify potential accidents in video frames by detecting and marking accident-related objects with bounding boxes. Further, this study [77,78] proposed the YOLO-NAS for the object detection model developed by Deci. It is pre-trained on extensive datasets such as COCO and Objects365, which enhances its precision in various tasks, including accident detection.

4. Conclusions

The early identification of potential accidents can enable faster reaction times and potentially prevent accidents. In the current research, the YOLOv8 model is employed to detect traffic accidents in metropolitan areas. The model is trained to recognize four categories: “car”,

“truck”, “bus”, and “accident.” Video streams from road surveillance cameras are collected as input data using Roboflow. The video frames are then preprocessed to enhance image quality and reduce noise through Deep-SORT. Finally, the YOLOv8 model is initialized with pre-trained weights obtained from a large dataset. The model is fine-tuned using the labeled dataset. After training, the model is capable of real-time accident detection. Results show that the accuracy (98.4%), recall (97.2%), precision (98.5%), and F1 score (95.7%) of the proposed approach are much greater than traditional methods. These outcomes highlight the potential of advanced machine learning to improve road safety and traffic management in metropolitan areas.

Author Contributions: Conceptualization, A.K. and M.A.R.; methodology, A.K. and M.A.R.; software, A.K. and M.A.R.; validation, Y.Z.A. and G.A.; formal analysis, Y.Z.A. and G.A.; investigation, Y.Z.A. and G.A.; resources, S.O. and M.S.H.; data curation, S.O. and M.S.H.; writing—original draft preparation, A.K. and M.A.R.; writing—review and editing, A.K. and M.A.R.; visualization, S.O. and M.S.H.; supervision, P.M. and M.S.H.; project administration, A.N. and P.M.; funding acquisition, G.A. and P.M. All authors have read and agreed to the published version of the manuscript.”

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Singh, S.K. Road traffic accidents in India: Issues and challenges. *Transp. Res. Procedia* **2017**, *25*, 4708–4719. [[CrossRef](#)]
2. Chand, A.; Jayesh, S.; Bhasi, A. Road traffic accidents: An overview of data sources, analysis techniques and contributing factors. *Mater. Today Proc.* **2021**, *47*, 5135–5141. [[CrossRef](#)]
3. Goniewicz, K.; Goniewicz, M.; Pawłowski, W.; Fiedor, P. Road accident rates: Strategies and programmes for improving road traffic safety. *Eur. J. Trauma Emerg. Surg.* **2016**, *42*, 433–438. [[CrossRef](#)]
4. Fontaras, G.; Zacharof, N.-G.; Ciuffo, B. Fuel consumption and CO₂ emissions from passenger cars in Europe—Laboratory versus real-world emissions. *Prog. Energy Combust. Sci.* **2017**, *60*, 97–131. [[CrossRef](#)]
5. Sacchi, R.; Bauer, C.; Cox, B.; Mutel, C. When, where and how can the electrification of passenger cars reduce greenhouse gas emissions? *Renew. Sustain. Energy Rev.* **2022**, *162*, 112475. [[CrossRef](#)]
6. Knobloch, F.; Hanssen, S.V.; Lam, A.; Pollitt, H.; Salas, P.; Chewpreecha, U.; Huijbregts, M.A.; Mercure, J.-F. Net emission reductions from electric cars and heat pumps in 59 world regions over time. *Nat. Sustain.* **2020**, *3*, 437–447. [[CrossRef](#)]
7. Yasin, Y.J.; Grivna, M.; Abu-Zidan, F.M. Global impact of COVID-19 pandemic on road traffic collisions. *World J. Emerg. Surg.* **2021**, *16*, 51. [[CrossRef](#)] [[PubMed](#)]
8. Muguro, J.K.; Sasaki, M.; Matsushita, K.; Njeri, W. Trend analysis and fatality causes in Kenyan roads: A review of road traffic accident data between 2015 and 2020. *Cogent Eng.* **2020**, *7*, 1797981. [[CrossRef](#)]
9. Shaik, M.E.; Islam, M.M.; Hossain, Q.S. A review on neural network techniques for the prediction of road traffic accident severity. *Asian Transp. Stud.* **2021**, *7*, 100040. [[CrossRef](#)]
10. Chang, F.-R.; Huang, H.-L.; Schwebel, D.C.; Chan, A.H.; Hu, G.-Q. Global road traffic injury statistics: Challenges, mechanisms and solutions. *Chin. J. Traumatol.* **2020**, *23*, 216–218. [[CrossRef](#)]
11. Khan, M.A.; Grivna, M.; Nauman, J.; Soteriades, E.S.; Cevik, A.A.; Hashim, M.J.; Govender, R.; Al Azezi, S.R. Global incidence and mortality patterns of pedestrian road traffic injuries by sociodemographic index, with forecasting: Findings from the global burden of diseases, injuries, and risk factors 2017 study. *Int. J. Environ. Res. Public Health* **2020**, *17*, 2135. [[CrossRef](#)] [[PubMed](#)]
12. Rajasekaran, R.B.; Rajasekaran, S.; Vaishya, R. The role of social advocacy in reducing road traffic accidents in India. *J. Clin. Orthop. Trauma* **2021**, *12*, 2–3. [[CrossRef](#)]
13. Handiso, A.; Mekebo, G.G.; Galdassa, A. Trends and determinants of road traffic accident human death in Kembata Tembaro zone, SNNPR, Ethiopia. *Sci. J. Appl. Math. Stat.* **2022**, *10*, 85–89.
14. Guevara, L.; Auat Cheein, F. The role of 5G technologies: Challenges in smart cities and intelligent transportation systems. *Sustainability* **2020**, *12*, 6469. [[CrossRef](#)]
15. Javed, A.R.; Shahzad, F.; ur Rehman, S.; Zikria, Y.B.; Razzak, I.; Jalil, Z.; Xu, G. Future smart cities: Requirements, emerging technologies, applications, challenges, and future aspects. *Cities* **2022**, *129*, 103794. [[CrossRef](#)]
16. Kumar, H.; Singh, M.K.; Gupta, M.; Madaan, J. Moving towards smart cities: Solutions that lead to the Smart City Transformation Framework. *Technol. Forecast. Soc. Chang.* **2020**, *153*, 119281. [[CrossRef](#)]
17. Gohar, A.; Nencioni, G. The role of 5G technologies in a smart city: The case for intelligent transportation system. *Sustainability* **2021**, *13*, 5188. [[CrossRef](#)]

18. Manfreda, A.; Ljubi, K.; Groznik, A. Autonomous vehicles in the smart city era: An empirical study of adoption factors important for millennials. *Int. J. Inf. Manag.* **2021**, *58*, 102050. [[CrossRef](#)]
19. Sarang, S.A.; Raza, M.A.; Panhwar, M.; Khan, M.; Abbas, G.; Touti, E.; Altamimi, A.; Wijaya, A.A. Maximizing solar power generation through conventional and digital MPPT techniques: A comparative analysis. *Sci. Rep.* **2024**, *14*, 8944. [[CrossRef](#)]
20. Lai, C.S.; Jia, Y.; Dong, Z.; Wang, D.; Tao, Y.; Lai, Q.H.; Wong, R.T.; Zobaa, A.F.; Wu, R.; Lai, L.L. A review of technical standards for smart cities. *Clean Technol.* **2020**, *2*, 290–310. [[CrossRef](#)]
21. Aljohani, M.; Olariu, S.; Alali, A.; Jain, S. A survey of parking solutions for smart cities. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10012–10029. [[CrossRef](#)]
22. Nikitas, A.; Michalakopoulou, K.; Njoya, E.T.; Karampatzakis, D. Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era. *Sustainability* **2020**, *12*, 2789. [[CrossRef](#)]
23. Mohamed, N.; Al-Jaroodi, J.; Jawhar, I.; Idries, A.; Mohammed, F. Unmanned aerial vehicles applications in future smart cities. *Technol. Forecast. Soc. Chang.* **2020**, *153*, 119293. [[CrossRef](#)]
24. Kashef, M.; Visvizi, A.; Troisi, O. Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. *Comput. Hum. Behav.* **2021**, *124*, 106923. [[CrossRef](#)]
25. Zhou, Q.; Zhu, M.; Qiao, Y.; Zhang, X.; Chen, J. Achieving resilience through smart cities? Evidence from China. *Habitat Int.* **2021**, *111*, 102348. [[CrossRef](#)]
26. Bhushan, B.; Khamparia, A.; Sagayam, K.M.; Sharma, S.K.; Ahad, M.A.; Debnath, N.C. Blockchain for smart cities: A review of architectures, integration trends and future research directions. *Sustain. Cities Soc.* **2020**, *61*, 102360. [[CrossRef](#)]
27. Kandt, J.; Batty, M. Smart cities, big data and urban policy: Towards urban analytics for the long run. *Cities* **2021**, *109*, 102992. [[CrossRef](#)]
28. Chu, J.; Zhang, C.; Yan, M.; Zhang, H.; Ge, T. TRD-YOLO: A real-time, high-performance small traffic sign detection algorithm. *Sensors* **2023**, *23*, 3871. [[CrossRef](#)]
29. Li, X.; Shi, B.; Nie, T.; Zhang, K.; Wang, W. Multi-object recognition method based on improved yolov2 model. *Inf. Technol. Control.* **2021**, *50*, 13–27.
30. Ayob, A.; Khairuddin, K.; Mustafah, Y.; Salisa, A.; Kadir, K. Analysis of pruned neural networks (MobileNetV2-YOLO v2) for underwater object detection. In Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19; Springer: Berlin/Heidelberg, Germany, 2021; pp. 87–98.
31. Huang, Y.-Q.; Zheng, J.-C.; Sun, S.-D.; Yang, C.-F.; Liu, J. Optimized YOLOv3 algorithm and its application in traffic flow detections. *Appl. Sci.* **2020**, *10*, 3079. [[CrossRef](#)]
32. Zhang, H.; Qin, L.; Li, J.; Guo, Y.; Zhou, Y.; Zhang, J.; Xu, Z. Real-time detection method for small traffic signs based on Yolov3. *IEEE Access* **2020**, *8*, 64145–64156. [[CrossRef](#)]
33. Dewi, C.; Chen, R.-C.; Liu, Y.-T.; Jiang, X.; Hartomo, K.D. Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. *IEEE Access* **2021**, *9*, 97228–97242. [[CrossRef](#)]
34. Dewi, C.; Chen, R.-C.; Jiang, X.; Yu, H. Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4. *Multimed. Tools Appl.* **2022**, *81*, 37821–37845. [[CrossRef](#)]
35. Wang, Q.; Zhang, Q.; Liang, X.; Wang, Y.; Zhou, C.; Mikulovich, V.I. Traffic lights detection and recognition method based on the improved YOLOv4 algorithm. *Sensors* **2021**, *22*, 200. [[CrossRef](#)]
36. Huang, Y.; Zhang, H. A safety vehicle detection mechanism based on YOLOv5. In Proceedings of the 2021 IEEE 6th international conference on smart cloud (SmartCloud), Newark, NJ, USA, 6–8 November 2021; pp. 1–6.
37. Murthy, J.S.; Siddesh, G.; Lai, W.-C.; Parameshachari, B.; Patil, S.N.; Hemalatha, K. Objectdetect: A real-time object detection framework for advanced driver assistant systems using yolov5. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9444360. [[CrossRef](#)]
38. Aboah, A. A vision-based system for traffic anomaly detection using deep learning and decision trees. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 4207–4212.
39. John, A.; Meva, D. Comparative Study of Various Algorithms for Vehicle Detection and Counting in Traffic. In Proceedings of the International Conference on Advancements in Smart Computing and Information Security, Rajkot, India, 24–26 November 2022; pp. 271–286.
40. Kaya, Ö.; Çodur, M.Y.; Mustafaraj, E. Automatic detection of pedestrian crosswalk with faster r-cnn and yolov7. *Buildings* **2023**, *13*, 1070. [[CrossRef](#)]
41. Li, S.; Wang, S.; Wang, P. A small object detection algorithm for traffic signs based on improved YOLOv7. *Sensors* **2023**, *23*, 7145. [[CrossRef](#)]
42. Balasundaram, A.; Shaik, A.; Prasad, A.; Pratheepan, Y. On-road obstacle detection in real time environment using an ensemble deep learning model. *Signal Image Video Process.* **2024**, *18*, 5387–5400. [[CrossRef](#)]
43. Pitts, H. Warehouse Robot Detection for Human Safety Using YOLOv8. In Proceedings of the SoutheastCon 2024, Atlanta, GA, USA, 15–24 March 2024; pp. 1184–1188.
44. Hou, X.; Wang, Y.; Chau, L.-P. Vehicle tracking using deep sort with low confidence track filtering. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6.
45. Liu, H.; Pei, Y.; Bei, Q.; Deng, L. Improved DeepSORT Algorithm Based on Multi-Feature Fusion. *Appl. Syst. Innov.* **2022**, *5*, 55. [[CrossRef](#)]

46. Gai, Y.; He, W.; Zhou, Z. Pedestrian target tracking based on DeepSORT with YOLOv5. In Proceedings of the 2021 2nd International Conference on Computer Engineering and Intelligent Control (ICCEIC), Chongqing, China, 12–14 November 2021; pp. 1–5.
47. Jie, Y.; Leonidas, L.; Mumtaz, F.; Ali, M. Ship detection and tracking in inland waterways using improved YOLOv3 and Deep SORT. *Symmetry* **2021**, *13*, 308. [\[CrossRef\]](#)
48. Wang, Y.; Zhang, K.; Wang, L.; Wu, L. An Improved YOLOv8 Algorithm for Rail Surface Defect Detection. *IEEE Access* **2024**, 44984–44997. [\[CrossRef\]](#)
49. Yang, S.; Wang, W.; Gao, S.; Deng, Z. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. *Comput. Electron. Agric.* **2023**, *215*, 108360. [\[CrossRef\]](#)
50. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [\[CrossRef\]](#)
51. Xiao, B.; Nguyen, M.; Yan, W.Q. Fruit ripeness identification using YOLOv8 model. *Multimed. Tools Appl.* **2024**, *83*, 28039–28056. [\[CrossRef\]](#)
52. Liu, L.; Li, P.; Wang, D.; Zhu, S. A wind turbine damage detection algorithm designed based on YOLOv8. *Appl. Soft Comput.* **2024**, *154*, 111364. [\[CrossRef\]](#)
53. Wen, Y.; Gao, X.; Luo, L.; Li, J. Improved YOLOv8-Based Target Precision Detection Algorithm for Train Wheel Tread Defects. *Sensors* **2024**, *24*, 3477. [\[CrossRef\]](#)
54. Wang, H.; Yang, H.; Chen, H.; Wang, J.; Zhou, X.; Xu, Y. A Remote Sensing Image Target Detection Algorithm Based on Improved YOLOv8. *Appl. Sci.* **2024**, *14*, 1557. [\[CrossRef\]](#)
55. Boudjit, K.; Ramzan, N. Human detection based on deep learning YOLO-v2 for real-time UAV applications. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 527–544. [\[CrossRef\]](#)
56. Saranya, K.C.; Thangavelu, A.; Chidambaram, A.; Arumugam, S.; Govindraj, S. Cyclist detection using tiny yolo v2. In *Soft Computing for Problem Solving: SocProS 2018*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 2, pp. 969–979.
57. Han, X.; Chang, J.; Wang, K. Real-time object detection based on YOLO-v2 for tiny vehicle object. *Procedia Comput. Sci.* **2021**, *183*, 61–72. [\[CrossRef\]](#)
58. Hou, X.; Zhang, Y.; Hou, J. Application of YOLO V2 in construction vehicle detection. In Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Xi'an, China, 1–3 August 2020; pp. 1249–1256.
59. Alsanad, H.R.; Ucan, O.N.; Ilyas, M.; Khan, A.U.R.; Bayat, O. Real-time fuel truck detection algorithm based on deep convolutional neural network. *IEEE Access* **2020**, *8*, 118808–118817. [\[CrossRef\]](#)
60. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2023**, *35*, 13895–13906. [\[CrossRef\]](#)
61. Kumar, S.; Gupta, H.; Yadav, D.; Ansari, I.A.; Verma, O.P. YOLOv4 algorithm for the real-time detection of fire and personal protective equipments at construction sites. *Multimed. Tools Appl.* **2022**, *81*, 22163–22183. [\[CrossRef\]](#)
62. Tan, L.; Lv, X.; Lian, X.; Wang, G. YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm. *Comput. Electr. Eng.* **2021**, *93*, 107261. [\[CrossRef\]](#)
63. Zeng, L.; Duan, X.; Pan, Y.; Deng, M. Research on the algorithm of helmet-wearing detection based on the optimized yolov4. *Vis. Comput.* **2023**, *39*, 2165–2175. [\[CrossRef\]](#)
64. Khan, M.; Raza, M.A.; Jumani, T.A.; Mirsaeidi, S.; Abbas, G.; Touti, E.-D.; Alshahir, A. Modeling of Intelligent Controllers for Solar Photovoltaic System Under Varying Irradiation Condition. *Front. Energy Res.* **2023**, *11*, 1288486. [\[CrossRef\]](#)
65. Dong, C.; Du, G. An enhanced real-time human pose estimation method based on modified YOLOv8 framework. *Sci. Rep.* **2024**, *14*, 8012. [\[CrossRef\]](#)
66. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. *Electronics* **2023**, *12*, 3664. [\[CrossRef\]](#)
67. Khan, M.; Aamir, M.; Hussain, A.; Badar, Y.; Sharif, M.; Faisal, M. Enhancing Solar Power Forecasting in Multi-Weather Conditions Using Deep Neural Networks. In Proceedings of the 2023 2nd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETEECTE), Lahore, Pakistan, 27–29 November 2023; pp. 1–11.
68. Zeng, Q.; Zhou, G.; Wan, L.; Wang, L.; Xuan, G.; Shao, Y. Detection of Coal and Gangue Based on Improved YOLOv8. *Sensors* **2024**, *24*, 1246. [\[CrossRef\]](#) [\[PubMed\]](#)
69. Lalinia, M.; Sahafi, A. Colorectal polyp detection in colonoscopy images using YOLO-V8 network. *Signal Image Video Process.* **2024**, *18*, 2047–2058. [\[CrossRef\]](#)
70. Ye, R.; Gao, Q.; Qian, Y.; Sun, J.; Li, T. Improved Yolov8 and Sahi Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea. *Agronomy* **2024**, *14*, 1034. [\[CrossRef\]](#)
71. Kumar, N.; Acharya, D.; Lohani, D. An IoT-based vehicle accident detection and classification system using sensor fusion. *IEEE Internet Things J.* **2020**, *8*, 869–880. [\[CrossRef\]](#)
72. Pillai, M.S.; Chaudhary, G.; Khari, M.; Crespo, R.G. Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning. *Soft Comput.* **2021**, *25*, 11929–11940. [\[CrossRef\]](#)
73. Huang, T.; Wang, S.; Sharma, A. Highway crash detection and risk estimation using deep learning. *Accid. Anal. Prev.* **2020**, *135*, 105392. [\[CrossRef\]](#) [\[PubMed\]](#)

74. Li, L.; Lin, Y.; Du, B.; Yang, F.; Ran, B. Real-time traffic incident detection based on a hybrid deep learning model. *Transp. A Transp. Sci.* **2022**, *18*, 78–98. [[CrossRef](#)]
75. Zaitouny, A.; Fragkou, A.D.; Stemler, T.; Walker, D.M.; Sun, Y.; Karakasidis, T.; Nathanail, E.; Small, M. Multiple sensors data integration for traffic incident detection using the quadrant scan. *Sensors* **2022**, *22*, 2933. [[CrossRef](#)]
76. Jaspin, K.; Bright, A.A.; Legin, M.L. Accident Detection and Severity Classification System using YOLO Model. In Proceedings of the 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 5–7 June 2024; pp. 1160–1167.
77. Nusari, A.N.M.; Ozbek, I.Y.; Oral, E.A. Automatic Vehicle Accident Detection and Classification from Images: A Comparison of YOLOv9 and YOLO-NAS Algorithms. In Proceedings of the 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Turkiye, 15–18 May 2024; pp. 1–4.
78. Chung, Y.L.; Lin, C.K. Application of a model that combines the YOLOv3 object detection algorithm and canny edge detection algorithm to detect highway accidents. *Symmetry* **2020**, *12*, 1875. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.