Scalable Machine Learning Model for Highway CCTV Feed Real-Time Car Accident and Damage Detection

Mhd Saeed Sharif
Computer Science and DT, ACE, UEL
University Way, London, UK
S.Sharif@uel.ac.uk

Abin Zorto
Computer Science and DT, ACE, UEL
University Way, London, UK
u2091940@uel.ac.uk

Victor Kamalu Brown
Computer Science and DT, ACE, UEL
University Way, London, UK
u2225990@uel.ac.uk

Wael Elmedany
College of Information Technology
University of Bahrain, Kingdom of Bahrain
welmedany@uob.edu.bh

***Abstract*—This study investigates the potential advantages of employing computer vision algorithms to enhance real-time accident detection and response on highways using CCTV feed. Traditional techniques rely on retrospective data, which can decrease response times and precision. Computer vision algorithms have the potential to enhance detection speed and precision, resulting in quicker emergency response and monitoring of traffic flow. The primary objective of this study is to identify the advantages of utilising computer vision algorithms and the data gathered through them to enhance road safety measures and reduce the occurrence of accidents. This study is anticipated to result in quicker emergency response times, the identification of areas where statistically more accidents are likely to occur, and the use of collected data for research purposes, which can lead to enhanced road safety measures. Using computer vision algorithms for accident detection and response has the potential to reduce the human and monetary costs associated with traffic accidents.**

***Index Terms*—Real-time accident detection, Computer vision, Object detection, Traffic monitoring, Road safety, Emergency response**

I. Introduction

Road incidents are a significant risk to human life and have a substantial economic impact globally. According to the World Health Organisation [1], each year, 1.35 million people are killed and 50 million are injured in road accidents. These accidents also result in economic losses estimated between 1 and 3 percent of a country's GDP [2]. Traditional accident detection methods rely on retrospective data analysis, leading to delayed response times and less accuracy [3].

The current method of accident detection relies on software programs like PC-CRASH and analysis of past accidents using data from police reports and eyewitness accounts [3]. This retrospective approach causes delays in alerting first responders and may lead to missed opportunities to save lives. Additionally, relying on human reporting introduces further delays and potential inaccuracies in the data collected [3].

To overcome these challenges and improve accident detection and response, computer vision algorithms, specifically object detection, have been proposed [4]. By analyzing real-time video feeds from traffic cameras using computer vision, accidents can be detected as they happen, enabling prompt alerts to first responders [4].

Using computer vision algorithms for accident detection offers several benefits. It enables real-time identification of incidents, leading to faster emergency response and potential life-saving interventions [4]. Additionally, computer vision can monitor traffic flow and identify accident-prone areas, allowing for targeted road safety measures [4]. Lastly, the data collected through computer vision provides valuable research material for identifying patterns and factors contributing to accidents, aiding in the development of evidence-based accident prevention strategies [4].

II. Literature Review

A. Sensors vs Machine Learning: A Comparison for Accident Detection

Haria et al. [5] propose a system based on ultrasonic and vibration sensors for collision prevention and detection. The system detects obstacles, reduces vehicle speed, and sends an SOS signal in case of a collision. However, it has limitations such as difficulty distinguishing between genuine obstacles and false positives and being ineffective in adverse weather conditions.

Almaadeed et al. [6] present an audio-based road hazard detection method using quadratic time-frequency distributions (QTFDs). Their method improves accuracy compared to other approaches, but lacks information about the dataset used and computational resource requirements, limiting its evaluation and practical feasibility.

Ohgushi et al. [7] propose dash-mounted vehicle cameras for collision identification but face the challenge of lengthy processing time for photographs. These studies demonstrate different approaches to accident detection, including machine learning, audio surveillance, and video analysis. However, they have limitations such as being limited to specific scenarios, not considering environmental factors, and using small datasets that may impact applicability to larger and diverse datasets.

Additionally, ethical implications of using machine learning models for accident detection, such as privacy concerns and potential misuse, are not thoroughly explored in these studies.

To advance the field, further research should address these limitations by using representative datasets, considering environmental factors, and carefully examining the ethical implications of implementing machine learning models for accident detection.

B. The Problem of Data Availability: Challenges and Solutions

Batanina et al. [8] address the challenge of limited crash video data by adapting video game scenes to CCTV footage. They use video games to generate car crash scenes for training a 3D CNN-based deep model. The study shows improved accuracy by leveraging synthetic video data, although there is a domain gap between synthetic and real videos that can impact the model's performance.

Maaloul et al. [9] propose a model for car accident detection focusing on motion detection, feature extraction, feature analysis, and accident recognition. Their model achieves a high recall rate and specializes in detecting roadside vehicle anomalies in a one-way traffic scenario. The model utilizes optical flow, noise, and statistical analysis to capture various traffic conditions and outperforms other methods.

Both studies contribute to car accident detection using different approaches. Batanina et al. [8] leverage synthetic data to enhance the model's performance on real-world footage, while Maaloul et al. [9] focus on motion detection and statistical analysis. These studies demonstrate the potential of machine learning techniques in improving car accident detection and highlight the importance of addressing challenges such as data availability and domain adaptation for real-world scenarios.

C. Enhancing Accuracy

Chen et al. [10] address the difficulties associated with traffic accident detection, including the construction of behavioural representations in congested environments and the localization of the accident region based on these representations. To address the first difficulty, the authors propose collision prediction by employing the target variable "the last second before collision" and introducing OF-SIFT, a temporal feature descriptor derived from optical flow. This method permits the construction of a more compact image representation capable of handling partial occlusion and independent of explicit shape structure.

Chen et al. [10] propose the use of the extreme learning machine (ELM) algorithm as the classifier for traffic accident detection for the second challenge. The ELM algorithm has an advantage over conventional methods like the Bayesian Probability Framework because it does not require prior knowledge about vehicle behaviour. This adaptability is a major asset of the ELM algorithm. In addition, prior research has demonstrated that ELM outperforms other classifiers in classification tasks [11] [12] [13].

Chen et al. [10] contribute to the enhancement of existing methods for traffic accident detection by resolving both obstacles. Their approach to collision prediction and use of the ELM algorithm as a classifier offer promising avenues for improving the precision of traffic accident detection systems.

D. Neural Networks: An Overview of Principles and Applications

Singh and Mohan [14] propose a framework for detecting road accidents using denoising autoencoders and unsupervised models. Their approach shows the superiority of convolutional autoencoders over hand-crafted features-based methods. However, the reliance on trajectory intersection points and the use of an unsupervised model may limit its applicability in certain situations.

Ijjin and Sharma [15] introduce a deformable deep convolutional neural network for generic object detection. Their pretraining strategy and the novel deformation-constrained pooling layer yield improved performance compared to previous methods. However, the selective search method for generating bounding boxes and potential inefficiency with large-scale datasets could be drawbacks.

Wu et al. [16] address the issue of gradient contribution imbalance in object detection training by proposing the gradient-balanced focal loss (GBFL). GBFL achieves a balance between foreground and background samples, leading to enhanced accuracy and convergence speed. Their findings demonstrate the superiority of GBFL over other state-of-the-art loss functions. However, the limitations of the proposed technique are not explicitly discussed.

Although these studies offer novel approaches to object detection and accident detection, there are limitations and scalability concerns that need to be addressed for practical application and scalability. Further research and investigation are required to optimize these methods and overcome their limitations in order to contribute to the development of more accurate and efficient object detection models.

E. Evaluating the Performance of Object Detection Models

Girshick et al. [17] developed a simple and scalable design for object detection, achieving a groundbreaking mean average precision (mAP) of 53.3%. They focused on object localization using a deep neural network and trained their models with annotated detection data. Their method, "R-CNN Regions with CNN Features," employed region proposals, feature extraction with a convolutional neural network, and classification with a linear support vector machine, resulting in precise localization. They also utilized a two-step training strategy to address the issue of insufficient training data, leading to improved accuracy.

Tan et al. [18] conducted a study comparing three object detection algorithms: Faster R-CNN, SSD, and YOLO. They created a dataset of oral solid dosage forms and evaluated the algorithms based on factors such as MAP score and detection speed. Faster R-CNN achieved the highest MAP score of 87.69 but had a slower detection speed of 7 FPS. YOLO had a faster detection speed of 51 FPS, making it suitable for real-time applications, while offering a satisfactory MAP score. SSD provided a moderate trade-off between detection speed and accuracy.

In conclusion, Girshick et al. [17] made significant contributions to object detection by improving mAP and developing efficient training strategies. Tan et al. [18] compared object detection algorithms and found that Faster R-CNN excelled in pinpointing objects with higher MAP scores, while YOLO performed well in real-time scenarios with faster detection speeds.

F. Human Detection

Human detection is a significant area in computer vision, and researchers have developed various methods to address the challenges associated with detecting humans in different poses, postures, sizes, colors, and shapes.

One notable method proposed by Dalal and Triggs [19] is the Histogram of Oriented Gradients (HOG) approach. They compared the performance of wavelets to HOG and found that HOG achieved better results. The HOG approach characterizes the outline of an object based on the distribution of local intensity gradients in an image. The image is divided into small regions called cells, and each cell forms a one-dimensional gradient histogram. The authors observed that larger cells, referred to as blocks, improved accuracy. These blocks are combined with a linear SVM classifier to create the human detection chain.

The HOG approach offers advantages by accurately capturing the edges of an image's shape. It incorporates fine orientation sampling and strong local photometric normalization, which handle variations in appearance and orientation. The HOG method has been widely employed in computer vision tasks such as face detection, pedestrian detection, and object recognition. Additionally, it has shown promising results in autonomous vehicles for detecting and tracking objects on the road.

G. Motor Lane Detection

Lane detection is a crucial application of object detection, with model-based and feature-based approaches being commonly used. Wang et al. [20] proposed a model-based algorithm based on the parabola model, while feature-based approaches involve extracting features like edges or lines and using techniques such as the Hough transform or Sobel edge detector.

The Hough transform shows promise in real-time lane detection but requires substantial computational power and can be affected by other objects in the image. The Sobel edge detector is a simpler and faster algorithm suitable for real-time applications but may be sensitive to noise and require additional preprocessing.

Ozgunalp and Kaymak [21] introduced a feature-based approach that reduces computational complexity by focusing on relevant image regions. Their method utilizes Canny edge detection and Hough transforms with geometric filtering to enhance accuracy.

Parajuli, Celenk, and Riley [22] proposed a method based on the vertical gradient of images, which is robust against horizontal shadows and does not require thresholding. This approach is immune to environmental and lighting conditions.

Both the feature-based approach by Ozgunalp and Kaymak [21] and the vertical gradient-based method by Parajuli, Celenk, and Riley [22] have their strengths and weaknesses in terms of accuracy, computational complexity, and robustness to environmental conditions.

To improve road safety, it is suggested to collect and annotate a large image dataset from various sources, preprocess the data, and train object recognition and classification models using the YOLO framework with the help of Roboflow. The models can be evaluated using accuracy metrics, considering weather and lighting conditions. However, ethical and bias concerns related to machine learning for car crash detection should be addressed, and further research is needed for accident detection and prevention to enhance road safety.

III. METHODOLOGY

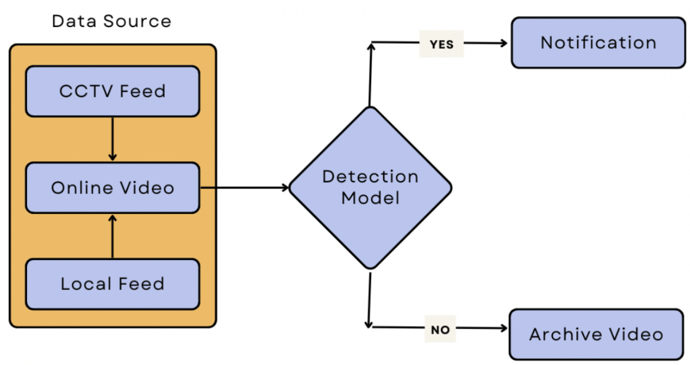

Fig. 1. Architecture of the proposed system

A. Convolutional Neural Network

Convolutional Neural Networks (CNNs) have revolutionized computer vision and image processing due to their exceptional performance in various applications. CNNs consist of different layers, including convolutional layers, pooling layers, and fully connected layers, which extract high-level features from input images using learned filters [23].

CNNs, also known as shift invariant or space invariant artificial neural networks, utilize convolution kernels or filters that slide across input features to generate feature maps. This shared-weight architecture allows CNNs to recognize features in different regions of the input image, making them efficient in processing complex image data [24].

Compared to other deep learning architectures, CNNs have several advantages. They require fewer parameters while achieving better results in tasks such as image and speech recognition [25]. CNNs can learn complex features from images, enabling applications in object detection, image processing, computer vision, and face recognition [26].

B. The YOLO (You Only Look Once) Algorithm

The YOLO (You Only Look Once) algorithm is a popular CNN object detection and image segmentation model in computer vision [27]. It follows a single-stage detection approach, directly predicting bounding boxes and class probabilities for all objects in an input image.

1) The YOLO (You Only Look Once) Architecture: The YOLO architecture consists of a backbone network for feature extraction, a detection head for prediction, and post-processing steps for removing duplicate detections using non-maximum suppression.

The working principle of YOLO involves passing an input image through the neural network to generate feature maps, which are used to predict bounding boxes and class probabilities. The YOLO model is simple and follows three steps: resizing the image, running a convolutional network, and thresholding the resulting detections [27].

The YOLO framework treats object detection as a regression problem, dividing the image into a grid and predicting bounding boxes, confidence scores, and class probabilities for each grid cell. It excels at understanding the overall context of the image and avoids mistaking background patches for objects [28].

C. Yolo v5 Model Architecture

1) Simple Pipeline: The YOLO model follows a refreshingly simple and straightforward pipeline, consisting of the following three steps:

1) Resizing the input image to 448 × 448.
2) Running a single convolutional network on the image.
3) Thresholding the resulting detections by the model's confidence.

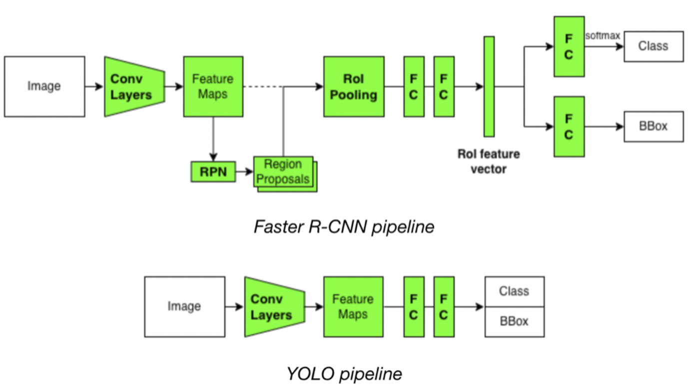

Fig. 2. YOLO simple pipeline. [29]

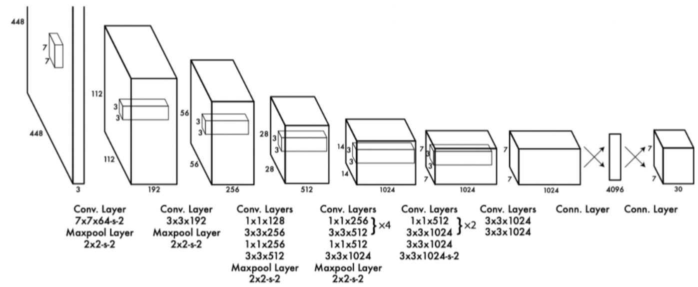

Fig. 3. YOLO Feature Extractor. [29]

2) YOLO Feature Extractor: Input images of size 448 × 448 (3 channels) are processed using YOLO's 24 convolutional layers, resulting in feature maps of size 7 × 7 with 1024 channels (see Figure 3).

In other words, YOLO divides the input image into 1024-dimensional vectors contained within a 7 × 7 grid.

3) YOLO Object Detection: YOLO treats object detection as a regression problem by dividing the image into an S × S grid. For each grid cell, it predicts B bounding boxes, confidence scores for those boxes, and C class probabilities. These predictions are encoded as an S × S × (B × 5 + C) tensor (see Figure 4).

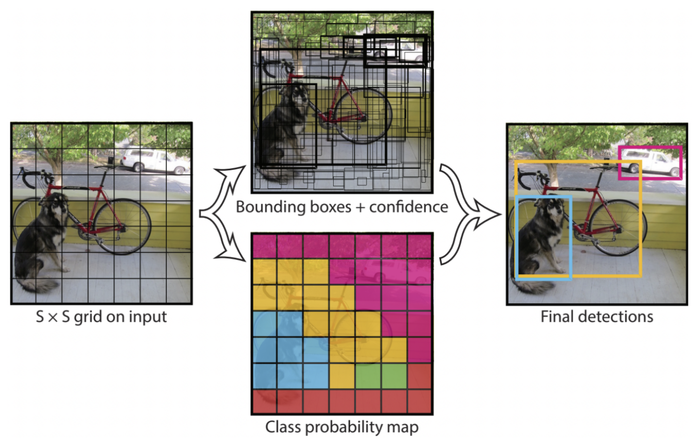

Fig. 4. YOLO Object Detection. [29]

This approach ensures that YOLO's detector understands the entire image and selects the right features, leading to fewer mistakes about the background compared to previous methods (see Figure 5).

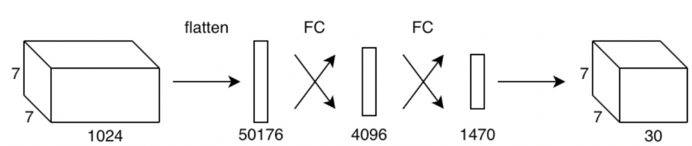

Fig. 5. Context understanding in YOLO Object Detection. [29]

The final fully connected layer generates a 1470-dimensional vector, which can be understood as a 7 × 7 × 30 tensor of predictions (see Figure 6).

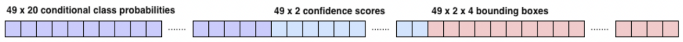

Fig. 6. YOLO Object Detection - 1470-dimensional vector layout. [29]

At first glance, the layout may seem confusing, but upon closer inspection, it consists of 49 different value sets, corresponding to the 49 grid cells (7×7) in the image. Each cell contains 30 values (1470 divided by 49), enabling YOLO to handle the 1470-dimensional vector as a $7 \times 7 \times 30$ tensor of predictions.

4) Limitations: While YOLO offers advantages in terms of speed and accuracy, it also has limitations. It may struggle with generalizing objects, localizing small objects, and imposing spatial constraints on bounding boxes [27]. YOLO also has a steep learning curve and may require high-end technology for real-time applications [30]. Researchers have proposed variations of YOLO to address these limitations.

D. Data-set Description

The dataset used in this investigation was sourced from the Roboflow Universe, which is a collection of open-source computer vision datasets and APIs (Roboflow). The dataset consists of 3,221 images categorized into accidents and damage. These images were annotated using the YOLO V5 PyTorch file format, with bounding boxes drawn around indications of accidents within each image. The dataset was divided into train, test, and validation folders (Roboflow).

1) Annotation Process: The annotation process was made more efficient with the use of the Roboflow annotation tool. This tool simplifies the addition of machine learning-related annotations to images by providing a user-friendly interface and annotation options such as bounding boxes, polygons, and semantic segmentation masks. The annotations were recorded in XML files compatible with popular deep learning frameworks like TensorFlow and PyTorch.

The Roboflow annotation tool offers several advantages, including ease of use, speed, flexibility, and accuracy. It streamlines the annotation process, allows for efficient annotation of multiple photos simultaneously, and provides various annotation options to meet specific requirements. The tool also includes features like zooming to focus annotations on specific image regions.

Bounding boxes and polygons are two commonly used annotation types in computer vision. Bounding boxes involve drawing rectangular boxes around objects or regions of interest, providing simplicity and productivity. However, they may lack accuracy and may not capture fine details accurately. On the other hand, polygons offer higher precision as they capture the shape and details of objects or regions more accurately. However, polygon annotation can be more time-consuming [29].

The choice between bounding boxes and polygons depends on the specific application's requirements and the desired level of precision. Bounding boxes may be suitable for simpler objects or regions, while polygons may be necessary for more complex ones [29].

E. Model Training Testing and Validation

In the training phase, the YOLO method utilizing convolutional neural networks (CNNs) was applied to train the model. The training data were stored in the ”training” folder on Google Drive, and the YOLO method allowed for fast image processing with a single forward pass.

After training, the model was tested using a separate dataset to evaluate its accuracy. The testing data were uploaded to the ”testing” folder on Google Drive, and OpenCV was utilized to process and test the model on this data. OpenCV is a popular computer vision library that offers various tools for image and video analysis. In this investigation, the trained model was applied to real-time car accident videos captured using OpenCV, enabling the identification and categorization of objects of interest.

Following the testing phase, the model underwent validation to ensure its accuracy. The validation data were transferred to the ”validating” folder on Google Drive, and the model's accuracy was evaluated using the Pandas and Matplotlib libraries. Pandas is a powerful tool for data analysis and manipulation, while Matplotlib provides visualization capabilities.

During the analysis phase, the accuracy score of the trained model was examined. The accuracy score represents the ratio of correct detections to the total number of detections. Matplotlib was employed to visualize the results of the analysis, offering a clear and concise representation of the accuracy score.

IV. RESULTS

The model was evaluated in terms of accuracy, precision, recall, and Area under the curve (AUC) metrics . The results of the experiments are visualized using tables, figures, and plots where appropriate.

A. Confusion Matrix

This confusion matrix illustrates the performance of a classifier trained to classify instances into three classes: ”accident,” ”damage,” and ”background.” The rows of the matrix represent the instances' actual class labels, while the columns represent the predicted class labels.

Fig. 7 shows the result of the confusion matrix, The classifier accurately identified 76% of instances belonging to the ”accident” class (true positives), indicating a relatively high accuracy in detecting accidents. However, there were 17% of instances from the ”accident” class that were misclassified as ”damage” (false negatives), suggesting some difficulty in distinguishing between these two classes. On the other hand, the classifier demonstrated good performance in recognizing instances from the ”background” class, with 78% of instances correctly classified as such (true negatives). However, there were some instances that were misclassified, with 21% of instances from the ”background” class being misclassified as

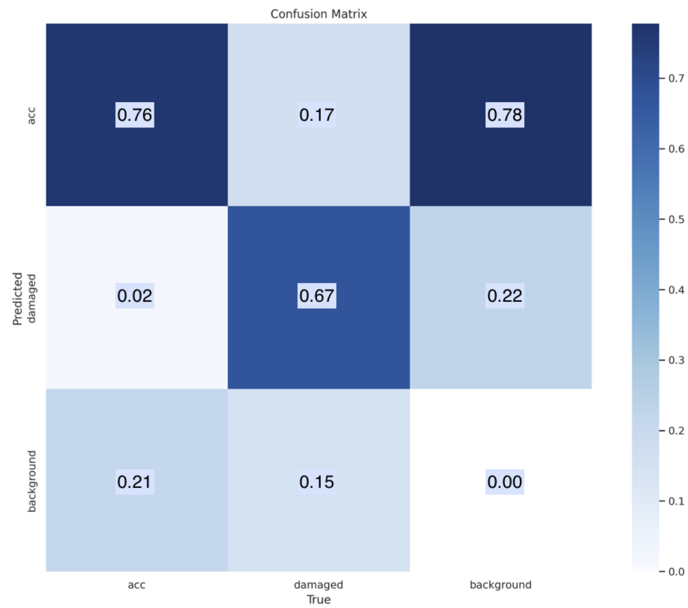

Fig. 7. Confusion Matrix

"accident" (false positives) and 15% as "damage" (false positives). Additionally, there were instances where the classifier showed confusion between the "damage" and "background" classes, with 2% of instances classified as "damage" actually belonging to the "accident" class (false positives) and 22% of instances classified as "damage" belonging to the "background" class (false positives). Overall, while the classifier demonstrated reasonable performance in detecting accidents and backgrounds, there is room for improvement in accurately distinguishing between accidents and damages, as well as properly classifying certain instances from the background class.

B. F1 Precision and Recall

The F1 score is a widely used metric for evaluating the performance of classification models. It combines precision and recall to provide a harmonic mean that assesses the model's ability to balance accurate positive predictions with capturing all actual positive cases. The F1 score helps determine the optimal confidence threshold for achieving a desired trade-off between precision and recall. In the provided figure, an F1 score of 0.67 suggests reasonable model performance with room for improvement. Precision is defined as the ratio of true positives to the sum of true positives and false positives, highlighting the model's accuracy in positive predictions. The precision-recall curve demonstrates the relationship between precision and recall at different classification thresholds. In the precision graph, recall is on the x-axis, precision on the y-axis, and accuracy is the positive predictive value. The graph shows increasing recall but decreasing precision, indicating the trade-off between accurately capturing positive cases and producing more outcomes. The precision-recall curve is particularly useful in object detection tasks, where object identification performance metrics such as mean average precision (mAP) are derived by averaging the maximum precision values at each recall level. The provided mAP value of 0.718 suggests accurate object detection, but there is room for improvement by adjusting the threshold. It is important to note that small threshold changes at precision-recall stages can significantly impact accuracy while slightly affecting recall.

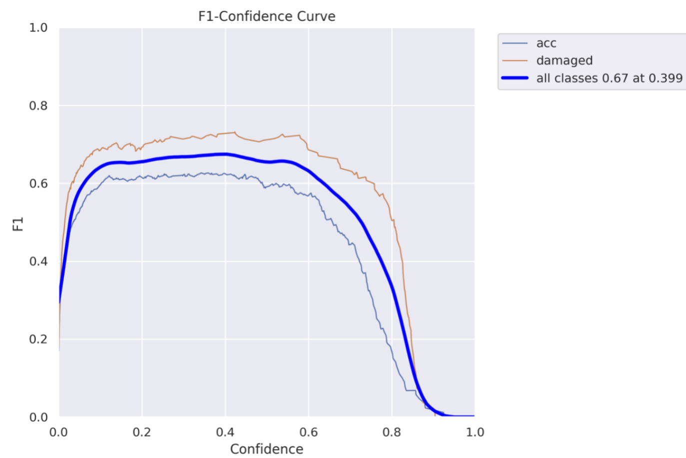

Fig. 8. F1 Confidence Curve

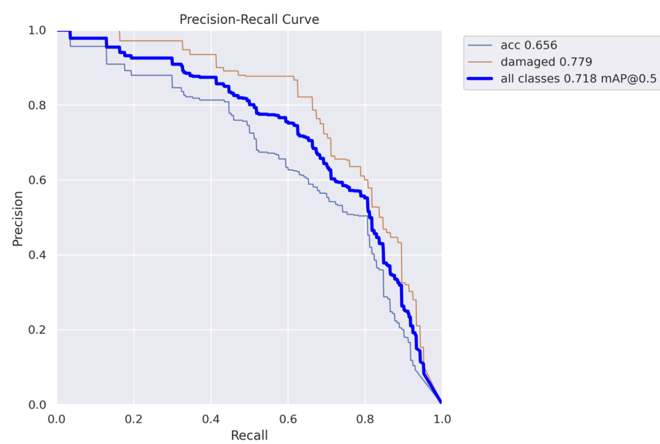

Fig. 9. Precision Recall Curve

C. Box Regression Loss (`train/box_loss`):

The box regression loss measures the model's ability to predict object location and size. A high box regression loss indicates difficulty in accurate predictions, while a decreasing loss value during training, from 1.6368 at epoch 1 to 0.79742 at epoch 100, indicates improved performance in predicting bounding box coordinates.

D. Classification Loss (`train/cls_loss`):

The classification loss evaluates the model's accuracy in predicting object class probabilities. A lower classification loss indicates improved performance in image classification. The loss decreases from 2.935 at epoch 1 to 0.64938 at epoch 99, showing enhancement in class probability predictions.

E. Deformable Convolution Layers (`train/dfl_loss`):

This loss measures the error in deformable convolution layers, which help detect objects of varying scales and aspect ratios. Lower dfl loss signifies better handling of object deformations and appearance variations. The loss decreases from 1.8467 to 1.1949, indicating improved performance in handling object variations.

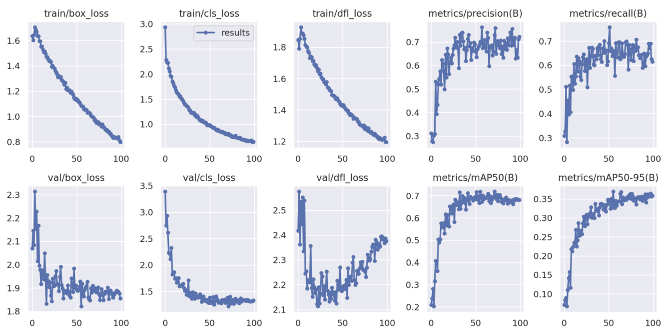

Fig. 10. Performance Metrics

F. Recall

Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify relevant instances of the target class. The recall value improves from 0.30843 at epoch 0 to 0.61539 at epoch 99, indicating increased detection of ground-truth objects. The false positive rate decreases from 69.16% to 38.46% over the epochs, suggesting improved object classification.

Algorithm	**Precision**	**Recall**	**F1**	**MAP**	**FPS**
YOLO v3	69.13	80.19	70.14	80.17	51
Faster R-CNN	62.19	94.24	78.23	87.69	7
SSD	63.17	88.69	72.13	82.41	32

V. Conclusion and Future Work

In conclusion, the developed accident detection model based on the YOLO framework demonstrates the capability to detect car accidents or damages in real time. The model has the potential to speed up emergency response and enhance traffic monitoring, leading to safer roads and reduced traffic accident costs. Additionally, the model can be applied to identify traffic flow patterns and trends, contributing to improved road design. The utilization of machine learning models for real-time accident detection can accelerate emergency response and mitigate the severity of injuries. Moreover, computer vision data can aid in identifying unsafe driving behaviors such as speeding and distracted driving, facilitating driver education and traffic enforcement.

In terms of future work, there are several areas for further development of the accident detection model. Firstly, efforts can be directed towards improving the model's ability to detect potential risks and more complex accidents. Training the model on video recordings of traffic, various types of vehicles, and road infrastructure can enhance its capabilities. This would enable the model to detect incidents in real time, resulting in quicker response times and potential lives saved.

Training the model on diverse traffic scenarios, including different types of vehicles such as trams, bikes, and trains, can enhance its accuracy and enable the identification of collisions involving multiple forms of transportation. Incorporating different road infrastructure elements like signs, poles, and roundabouts can provide a deeper understanding of accident causation factors and enhance accident detection. Moreover, training the model on datasets depicting close proximity between vehicles can enable the identification of potential dangers before they escalate into actual accidents, contributing to overall road safety.

Expanding the model's recognition capabilities to include pedestrians and cyclists in addition to vehicles can further enhance safety measures and improve accident detection involving vulnerable road users. Additionally, training the model to recognize crashes involving multiple vehicles can provide a more comprehensive picture of road safety and enable the detection of more complex events.

In application, the accident detection model can be utilized in various contexts, including traffic management systems, public transit services, and private vehicles. It can aid in real-time traffic flow monitoring and identification of incidents that may cause congestion or delays in traffic management systems. Public transport services can benefit from the model by detecting accidents on their routes and notifying drivers to take alternative routes. Integrating the model into dashcams and onboard systems of private vehicles can provide collision detection warnings to drivers or alert emergency services. Moreover, smart cities can leverage the accident detection model to monitor accident-prone areas and inform city officials for infrastructure improvements such as speed cameras or street lighting. The model can also contribute to workplace safety by analyzing camera feeds from manufacturing and construction sites to detect accidents and notify safety officers or emergency services.

In summary, the accident detection model demonstrates its applicability in various settings and can contribute to real-time accident detection, improved road safety, efficient traffic flow, and accident research. Further advancements and applications of the model have the potential to save lives, enhance overall road safety, and reduce the impact of accidents.

References

[1] World Health Organisation. Global status report on road safety 2018. https://www.who.int/publications/i/item/9789241565684, June 2018. Accessed: 2023-7-11.

[2] Pendela Kanchanamala, Ramanathan Lakshmanan, B. Muthu Kumar, and Balajee Maram. Aaco: Aquila anti-coronavirus optimization-based deep lstm network for road accident and severity detection, Apr 2023.

[3] Siqi Liu, Donghao Zhang, Yang Song, Hanchuan Peng, and Weidong Cai. Automated 3-d neuron tracing with precise branch erasing and confidence controlled back tracking. *IEEE Transactions on Medical Imaging*, 37(11):2441–2452, Nov 2018.

[4] Christopher Thompson, Jules White, Brian Dougherty, Adam Albright, and Douglas C. Schmidt. Using smartphones to detect car accidents and provide situational awareness to emergency responders, Jan 2010.

[5] Sahil Haria, Shubham Anchaliya, Vaibhav Gala, and Tina Maru. Car crash prevention and detection system using sensors and smart poles. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 800–804, 2018.

[6] Noor Almaadeed, Muhammad Asim, Somaya Al-Maadeed, Ahmed Bouridane, and Azeddine Beghdadi. Automatic detection and classification of audio events for road surveillance applications. *Sensors*, 18(6):1858, Jun 2018.

[7] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road obstacle detection method based on an autoencoder with semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[8] Elizaveta Batanina, Imad Eddine Ibrahim Bekkouch, Youssef Youssry, Adil Khan, Asad Masood Khattak, and Mikhail Bortnikov. Domain adaptation for car accident detection in videos. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2019.

[9] Boutheina Maaloul, Abdelmalik Taleb-Ahmed, Smail Niar, Naim Harb, and Carlos Valderrama. Adaptive video-based algorithm for accident detection on highways. In *2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–6, 2017.

[10] Yu Chen, Yuanlong Yu, and Ting Li. A vision based traffic accident detection method using extreme learning machine. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 567–572, 2016.

[11] Mahesh Pal. Extreme-learning-machine-based land cover classification. *International Journal of Remote Sensing*, 30(14):3835–3841, 2009.

[12] Songyun Xie, You Wu, Yunpeng Zhang, Juanli Zhang, and Chang Liu. Single channel single trial p300 detection using extreme learning machine: Compared with bpnn and svm. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 544–549. IEEE, 2014.

[13] Guang-Bin Huang, Dian Hui Wang, and Yuan Lan. Extreme learning machines: a survey. *International journal of machine learning and cybernetics*, 2:107–122, 2011.

[14] Dinesh Singh and Chalavadi Krishna Mohan. Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):879–887, 2018.

[15] Earnest Paul Ijjina and Sanjay Kumar Sharma. Accident detection from dashboard camera video. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4. IEEE, 2019.

[16] Zhihao Wu, Chengliang Liu, Chao Huang, Jie Wen, and Yong Xu. Deep object detection with example attribute based prediction modulation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2020–2024. IEEE, 2022.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[18] Lu Tan, Tianran Huangfu, Liyao Wu, and Wenying Chen. Comparison of retinanet, ssd, and yolo v3 for real-time pill identification. *BMC medical informatics and decision making*, 21:1–11, 2021.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[20] Bo Wang, Quan Chen, Min Zhou, Zhiqiang Zhang, Xiaogang Jin, and Kun Gai. Progressive feature polishing network for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12128–12135, 2020.

[21] Umar Ozgunalp and Sertan Kaymak. Lane detection by estimating and using restricted search space in hough domain. *Procedia computer science*, 120:148–155, 2017.

[22] Avishek Parajuli, Mehmet Celenk, H Bryan Riley, et al. Robust lane detection in shadows and low illumination conditions using local gradient features. *Open Journal of Applied Sciences*, 3(01):68, 2013.

[23] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53:5455–5516, 2020.

[24] Lijisha T and Ramitha MA. Multi-class cardiac diagnostic decision support system based on phonocardiogram signal. *international journal of engineering technology and management sciences*, 6(5):798–811, September 2022.

[25] Fudong Cai, Xiaobin Sun, Xiaoqing Liu, Jie Yang, Guoxin Guo, Zhiqiang Kong, Huanyun Liu, and Changfeng Lv. Research on convolutional neural network method for transmission line abnormal sound identification. In *Fifth International Conference on Mechatronics and Computer Technology Engineering (MCTE 2022)*, volume 12500, pages 1358–1361. SPIE, 2022.

[26] Mohammad Mustafa Taye. Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions. *Computation*, 11(3):52, 2023.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[28] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[29] Naoki and Naoki. Yolo: You look only once (the 1st version) - kikaben, Jul 2023.

[30] Wei Fang, Lin Wang, and Peiming Ren. Tinier-yolo: A real-time object detection method for constrained environments. *IEEE Access*, 8:1935–1944, 2019.