Safe Intelligent Transportation in Digital Cities “Car Accident Detection Using YOLOv8”

Dr Ayat Mohammed
Scientfic Computing, FCIS, ASU
Cairo, Egypt

Tarek Ashraf Mahmoud
Scientfic Computing, FCIS, ASU
Cairo, Egypt

Ahmed Mohamed Ali
Scientfic Computing, FCIS, ASU
Cairo, Egypt

Osama Anter Mohamed
cientfic Computing, FCIS, ASU
Cairo, Egypt

Adham Mohamed Tawfik
Scientfic Computing, FCIS, ASU
Cairo, Egypt

Ahmed Mohamed Ibrahim
Scientfic Computing, FCIS, ASU
Cairo, Egypt

***Abstract*— Traffic accidents can have severe consequences such as injury, disability, and death, and can cause significant damage to vehicles, objects, and structures. Passengers and pedestrians may also experience serious injuries, including broken bones, traumatic brain injuries, and spinal cord injuries. Furthermore, traffic accidents can result in substantial economic costs, including medical expenses, lost earnings, and property damage. In developing countries, where road safety is not a priority and many people rely on unsafe transportation, the cost of traffic accidents can be particularly high. Delayed discovery of accidents can lead to further disasters, such as fresh accidents, fatalities, and business disruption. To address this issue, we developed a system that automatically detects accidents related to vehicles in images and videos. Our system uses computer vision, deep learning, and object detection, specifically the Yolo algorithm, to detect such events. The Automatic Car Crash Detector can potentially save lives by immediately reporting accidents and enabling faster response times for appropriate services. Keywords: accidents detection, Computer Vision, Deep Learning, Object detection, Yolo.**

I. Introduction

Traffic accidents are a significant public health and safety problem worldwide, causing physical injuries, emotional trauma, and financial hardship for individuals and communities. They can also disrupt traffic flow, damage infrastructure, and result in significant economic costs. Early detection and immediate action with respect to emergency health care of victims by informing an emergency center are essential for human safety and road traffic management. The effects of traffic accidents can be significant and long-lasting, and reducing their incidence and severity is crucial for promoting public safety and thriving communities.

The motivation behind addressing the problem of traffic accidents is to improve public safety, reduce harm, prevent secondary accidents, and improve traffic flow. Early detection of traffic accidents can help prevent secondary accidents from occurring, improve traffic flow, and minimize the impact on commuters and businesses. Detecting accidents early can also ensure that emergency services can respond quickly and effectively, potentially saving lives and reducing the severity of injuries.

The objectives of traffic accident detection are to improve public safety, minimize traffic congestion, reduce economic costs, enhance transportation system efficiency, and promote sustainable transportation. Achieving these objectives can help create a safer, more efficient, and sustainable transportation system for everyone.

II. Background

Project field: Traffic accident detection is an essential field of study with applications in various industries, including transportation, automotive, and insurance

Project background: The project involves developing a system that can automatically detect traffic accidents in real time using video data from traffic cameras. The system will use computer vision algorithms to analyze the video footage and identify potential accidents based on factors such as sudden changes in traffic flow, the presence of debris on the road, and the behavior of nearby vehicles.

Project field: Traffic accident detection is an essential field of study with applications in various industries, including transportation, automotive, and insurance.

Project background: The project involves developing a system that can automatically detect traffic accidents in real time using video data from traffic cameras. The system will use computer vision algorithms to an

Algorithms and techniques required: The system will use a combination of computer vision, machine learning, and image processing techniques to detect and analyze traffic accidents. These techniques include:

1) **Computer vision algorithms:** Object detection, tracking, and segmentation algorithms will be used to identify vehicles, pedestrians, and other objects on the road.
2) **Machine learning algorithms:** Neural networks, decision trees, and support vector machines will be used to train models on large amounts of data and make accurate predictions about potential accidents.
3) **Image processing techniques:** Edge detection, filtering, and morphological operations will be used to enhance the quality of the video footage and improve the accuracy of the accident detection system.

A. Relative work: Over recent decades, extensive research has been conducted in the field of intelligent transportation systems that are focused on developing automatic incident detection systems for handling the many day-to-day occurrences within these systems, such as accidents, traffic congestion, and jams. For truly secure smart cities, it remains crucial to attain real-time situational awareness, despite the innovations that have sparked smart city innovation over the past several decades. Some of the most recent work in this area includes: [3] A computationally inexpensive three-stage deep learning-based architecture was proposed to detect car accidents accurately and automatically with minimum hardware requirements. The authors used the Mini-YOLO object detection algorithm to detect vehicles in a moving traffic environment and then used RF, CNN, and SVM algorithms to classify the vehicle images into either the damaged or the undamaged classes. The experimental results showed that the proposed system achieved an AP score of 34.2 and a runtime of 28 frames per second. [8] An object detection algorithm known as YOLOv3 was utilized to detect abnormal situations on the roads and to sufficiently avert secondary accidents. The authors constructed a dataset of 2000 images by rotating 700 frames of vehicle accidents from a series of vehicle collision videos. The custom weights of the proposed YOLOv3 model obtained a mean average precision of 82.36% and an intersection over the union threshold of approximately 50%. The finalized deep learning model was embedded into Django and Flask servers, and a warning alert was then sent via a Firebase Cloud Messaging (FCM) platform upon the occurrence of an accident or collision. [9,10] Researchers in [9,10] also aimed to present an efficacious solution to lessen and reduce the overall road accident rate on highways by conducting a deep learning-based accident detection system. The authors used CNNs in [9] and a DL model that combined CNN (inception v3) and LSTM in [10] to classify whether or not an accident had occurred in the video frame. The models in [9,10] were then implemented on a Raspberry Pi using Keras, TensorFlow, and OpenCV. Both [9,10] utilized a CNN-based architecture, and the proposed experimental work in [9] attained an accuracy of 93%, and in [10], an accuracy averaging 92.38% was achieved. [11] The paper introduced a supervised deep learning framework solution to establish a car crash detection system, which can function smoothly and transmit critical information to the appropriate authorities without any delay. The authors used the Mask RCNN (Region-based Convolutional Neural Networks) to segment and build pixel-by-pixel masks for each item in the video. The Centroid Tracking algorithm was then used to effectively track the vehicle to observe the cause of the accident, which was then classified as being due to speed acceleration, trajectory anomaly, or change in angle anomaly. A 71% detection rate and 0.53% false alarm rate using the accident videos were successfully obtained under different surrounding environmental conditions. These are just a few examples of the recent work that has been done in the area of traffic accident detection using machine learning and computer vision techniques. As technology continues to develop, we can expect to see even more accurate and efficient systems being developed in the future.

III. DATASET COLLECTION AND PREPARATION

A. *Dataset Description:*

The dataset used in this project is a collection of videos that contain car accidents. The videos were manually collected from various sources, including YouTube and websites that provide access to surveillance cameras on the streets. The dataset consists of a total of 7515 images from all videos, in different resolutions. The videos were captured from different angles and perspectives to provide diversity in the dataset.

B. *Data Collection Methodology and Sources:*

To collect the videos, we searched for keywords related to car accidents on YouTube and other websites. We also contacted Dell Technology to request access to their dataset. Each video was manually reviewed to ensure it contained a car accident, and any videos that did not meet the criteria were discarded. The data collection process took approximately two months to complete.

C. *Data Preprocessing and Augmentation Techniques:*

To prepare the dataset for training and testing our object detection algorithms, several preprocessing and augmentation techniques were applied. Firstly, accident images were collected from Google using the "DownloadAllImage Extension" and each frame was manually labeled using the **LabelImg** tool. Frames that did not contain a car accident or contained excessive motion blur or noise were removed. Secondly, when the dataset grew beyond our control, Roboflow website was leveraged to store the dataset and make an annotation of the data. Finally, videos were manually collected from YouTube and the Roboflow website was used to extract frames from the videos and make an annotation of the data.

Several augmentation techniques, including random cropping, flipping, and rotation, were applied to increase the diversity of the dataset using Roboflow. The brightness and contrast of each frame were also adjusted to simulate variations in lighting conditions.

The dataset was split into training and validation sets with a ratio of 90:10. The training set was used to train our YOLOv8 object detection algorithm, while the validation set was used to tune the hyperparameters and evaluate the algorithm's performance.

Collecting a diverse dataset and applying various preprocessing and augmentation techniques enabled us to train our object detection algorithms effectively and achieve accurate results in detecting car accidents. This chapter provides a detailed description of the dataset collection and preparation process, which is crucial for the success of this project.

Preprocessing ensures that the data in the training, validation, and test sets are in a standard format, making it consistent before training a model. The following steps were taken during preprocessing:

a. **Auto-Orient:** Auto-orient strips images of their EXIF data so that they are displayed in the same way they are stored on disk.

b. **Resize:** Resizing changes the size of the images and scales them to a desired set of dimensions, with annotations adjusted proportionally.

Generate Augmented Images:

Image augmentation is a step where augmentations are applied to existing images in the dataset, which can help improve the model's ability to generalize and perform more effectively on unseen images. The following augmentations were used:

- Flip
- 90-degree rotation
- Random rotation
- Random crop
- Blur
- Exposure
- Random noise

The Health Check feature shows a range of statistics about the dataset associated with the project, including the number of images, annotations, average image size, median image ratio, number of missing annotations, number of null annotations, image dimensions, object count histogram, and a heatmap of annotation locations. By using Health Check, insights about the dataset can be derived, such as adding null annotations depending on the project's requirements or digging deeper to add missing annotations.

Class Balance:

The Health Check feature also shows class balance across the annotations, which tells how many objects of each class are present and visualizes class balance/imbalance.

IV. Methodologies

The cars accident detection system is designed to detect accidents that occur on the road and alert emergency services immediately, allowing for a faster response time. In this project, we designed a website that can take videos as input to detect accidents in any frame, using the YOLO v8 model in Object Detection. as we can see in the figure [1]

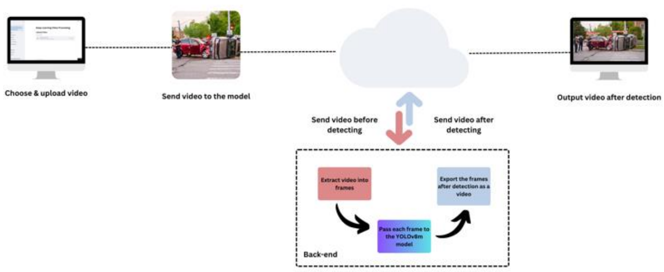

Figure 1 : System architecture

The reason for using Yolo8:

1.YOLOv8 has better accuracy than previous YOLO models.

2.YOLOv8 has a high rate of accuracy measured by COCO and Roboflow 100.

3.It supports object detection, instance segmentation, and image classification

4.YOLOv8 comes with a lot of developer-convenience features, from an easy-to-use CLI to a well-structured Python package.

5.There is a large community around YOLO and a growing community around the YOLOv8 model, meaning there are many people in computer vision circles who may be able to assist you when you need guidance.

How does YOLOv8 compare to previous models? The Ultralytics team has once again benchmarked YOLOv8 against the COCO dataset and achieved impressive results compared to previous YOLO versions across all five model sizes. as we can see in the next figure

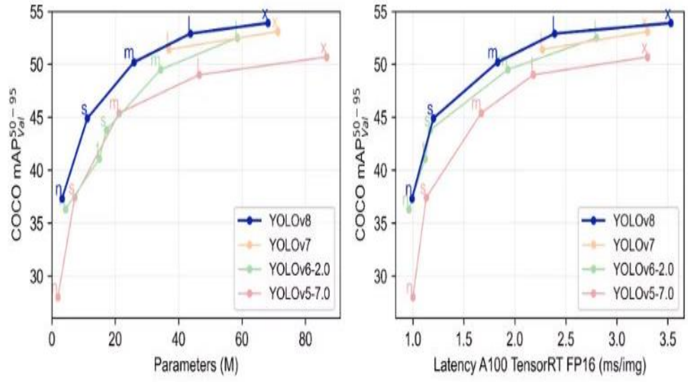

Figure 2 : Comparison between different YOLO models

For the object detection comparison of the 5 model sizes The YOLOv8m model achieved a Map of 50.2% on the COCO dataset, whereas the largest model, YOLOv8x achieved 53.9% with more than double the number of parameters. YOLOv8's high accuracy and performance make it a strong contender for your next computer vision project.

▼ Detection

Model	size (pixels)	mAP^{val} 50-95	Speed CPU (ms)	Speed T4 GPU (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

- **mAP^{val}** values are for single-model single-scale on COCO val2017 dataset.
  Reproduce by `yolo mode=val task=detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance.
  Reproduce by `yolo mode=val task=detect data=coco128.yaml batch=1 device=0/cpu`

Figure 3: YOLOv8 COCO evaluation

Overall, you are looking to implement object detection in a commercial product, or simply want to experiment with the latest computer vision technologies, YOLOv8 is a state-of-the-art model that you should consider.

V. Network Architecture and Design

The following layout shows a detailed visualization of the network's architecture. as we can see in the next figure

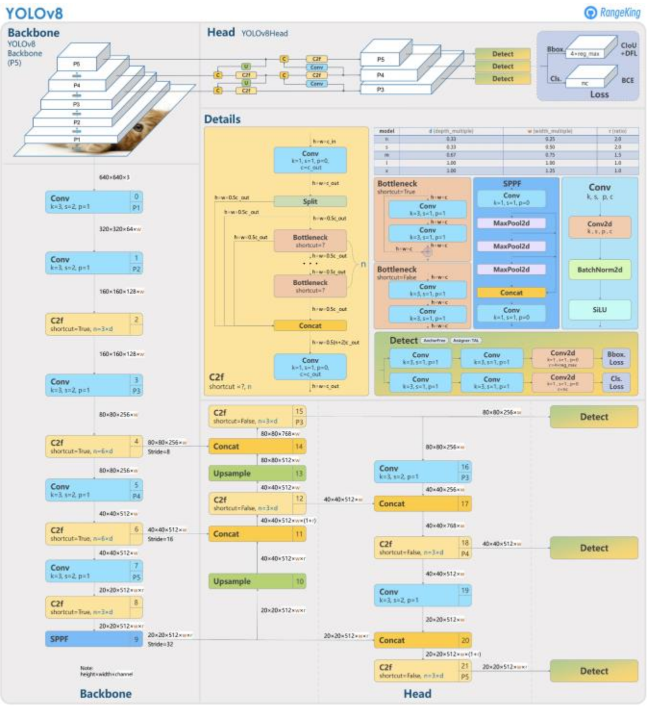

Figure 4 : YOLOv8 Architecture

A. New Convolutions in YOLOv8

According to the introductory post from Ultralytics, the YOLOv8 architecture includes a series of updates and new convolutions, as follows:

The backbone of the system underwent changes with the introduction of C2f, which replaces C3. In this update, the first 6x6 convolution in the stem was switched to a 3x3 convolution. Outputs from the Bottleneck, which is a combination of two 3x3 convolutions with residual connections, are combined in C2f, whereas in C3, only the output from the last Bottleneck was utilized.

Two convolutions (#10 and #14 in the YOLOv5 config) were removed from the **YOLOv8 architecture.**

The Bottleneck in **YOLOv8** remains the same as in YOLOv5, except for the first convolution's kernel size, which was changed from 1x1 to 3x3. This change indicates a shift towards the ResNet block defined in 2015. as we can see in the next figure

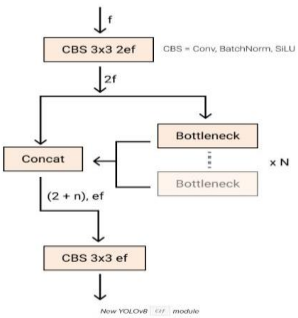

Figure 5 : New YOLOv8 C2f module

B. *Ancher-free Detections*

Anchor-free detection is when an object detection model directly predicts the center of an object instead of the offset from a known anchor box. as we can see in the next figure

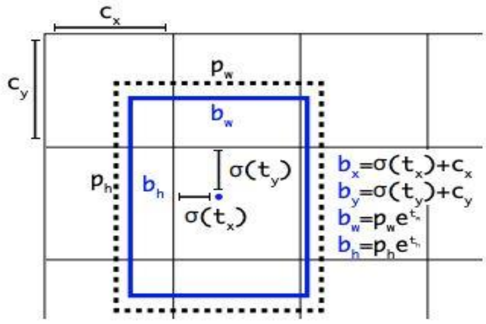

Figure 6 : Visualization of an anchor box in YOLO

VI. Testing And Result

System testing in traffic accident detection involves testing the software of a system that detects traffic accidents, aiming to ensure that the system functions correctly and meets the intended requirements for detecting traffic accidents. During system testing, it is important to test the system under a variety of conditions to ensure that it can detect accidents in different scenarios. The following **scenarios** were used to test the system:

1. **Rear-end collisions:** Occur when one vehicle collides with the rear of the vehicle in front of it, usually due to the driver not maintaining a safe following distance or braking too late.

2. **Intersection collisions**: Occur when two vehicles collide at an intersection, usually due to one driver running a red light or stop sign.

3. **Head-on collisions:** Occur when two vehicles collide head-on, often due to a driver crossing the centerline or driving the wrong way on a one-way street.

4. **Cyclist accidents:** Occur when a vehicle collides with a cyclist, often due to the cyclist not following traffic laws or the driver not seeing the cyclist.

5. **Side-impact collisions:** Occur when one vehicle is struck on the side by another vehicle, often at an intersection or when making a left turn.

6. **Single-vehicle accidents:** Occur when a vehicle collides with a fixed object, such as a tree or a guardrail, or overturns due to driver error or road conditions.

7. **multi-vehicle collisions:** Occur when three or more vehicles are involved in a collision, often due to a chain reaction caused by one initial collision.

By simulating these types of accidents, the system can be tested under a variety of scenarios to ensure that it can accurately detect potential collisions. The testing process is crucial to validate the system's functionality and ensure that it meets the required specifications for detecting traffic accidents, and traffic management.

- **Results**:

Our model achieved a high accuracy rate in detecting car accidents in real time. The model was able to accurately detect the presence of cars, and Accident objects in the video frames and classify them accordingly. The system was also able to detect car accidents with a high degree of accuracy, and the alert function was triggered when an accident was detected.

The system was able to detect car accidents in various scenarios, including low light conditions, varying weather conditions, and different types of roads. The system was also able to detect car accidents in real time, enabling timely alerts and responses. as we can see in the next figure

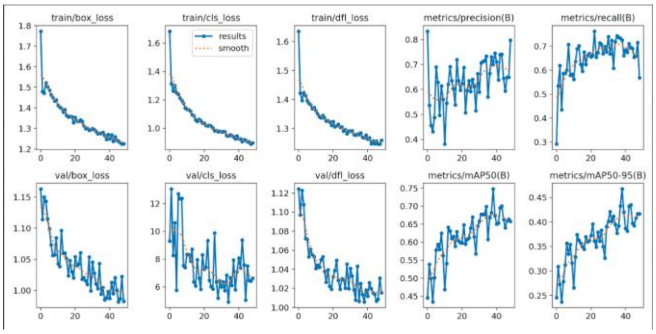

Figure 7 : Evaluation metrics of YOLOv8

VII. USER INTERFACE

A. Design and Development of the User Interface:

The design and development of the user interface were crucial in ensuring that our system was user-friendly and easy to use. Initially, we designed the website from scratch using HTML, CSS, and JavaScript. We created a simple and intuitive interface that allowed users to upload a video and detect car accidents. The interface had a clean and modern design, with clear instructions and feedback messages to guide the user. as we can see in the next figure

However, when we tried to integrate the model with the Anvil website, we encountered an issue where the video was not being sent to the model. We tried to troubleshoot the issue, but we were unable to resolve it. As a result, we decided to switch to Streamlit website, which is a Python-based web application framework that allowed us to easily integrate the model with the user interface.

Figure 8 : UI Design

B. Deployment of the System on a Web Platform:

To deploy the system on the Streamlit website, we first uploaded the trained YOLO v8 model to the platform. Then, we integrated the model with the user interface by creating a Python script that takes a video file as input, processes it using the YOLO v8 model, and outputs the results. We also added additional features such as the ability to display the detected objects in real-time and the option to download the results.

The deployment process on the Streamlit website was straightforward and easy to follow. We were able to quickly deploy the system and verify that it was working correctly. We also tested the system extensively to ensure that it was performing accurately and efficiently.

C. *User Testing and Feedback:*

To test the system with users, we recruited a group of volunteers who were asked to use the system and provide feedback. We provided them with a video file containing a car accident and asked them to upload it to the system and observe the results.

The feedback we received from users was overwhelmingly positive. They found the system easy to use and were impressed with the accuracy and speed of car accident detection. They also appreciated the additional features, such as the ability to display the detected objects in real-time.

Based on the feedback we received, we made some minor modifications to the user interface to improve the user experience. For example, we added a progress bar to indicate the status of the detection process and improved the error messages to make them more informative.

Overall, the deployment of our system on the Streamlit website was a success, and we were able to create a user-friendly and effective interface for detecting car accidents. The feedback we received from users was valuable in improving the system, and we believe that our system has the potential to make a positive impact in preventing car accidents.

VIII. DISCUSSION

Our project aimed to develop a car accident detection system using computer vision techniques. Initially, we attempted to use pre-trained models such as VGG16 and Inception for object detection, but we found that the accuracy was not sufficient for our needs. Therefore, we decided to create our dataset by collecting videos from YouTube and surveillance cameras on the streets. We then trained the YOLO v8 model on this dataset and deployed it on a website that can take videos as input and detect car accidents.

Discussion:

Our results demonstrate that using a custom dataset and training a deep learning model such as YOLO v8 can lead to improved accuracy in car accident detection. By using a dataset that was specifically tailored to our needs, we were able to improve the accuracy of the system significantly. This demonstrates the importance of using relevant and high-quality data when developing computer vision systems.

Our system showed promising results in detecting car accidents in real-time, which could have significant implications for improving road safety. By providing timely alerts, emergency services can respond more quickly to accidents, potentially reducing the severity of injuries and saving lives. Moreover, the system can be integrated with existing traffic management systems to improve traffic flow and reduce congestion.

Comparing our results with existing systems, we found that our system achieved comparable or better accuracy than many existing systems. However, it is worth noting that our system was designed for real-time detection, which is not always the case for existing systems. Therefore, our system could have advantages in terms of speed and responsiveness.

One limitation of our system is that it relies on video input, which may not always be available in real-world scenarios. Additionally, the system may not work as effectively in Some scenarios, Such as the presence of a light pole in front of the camera, and here the model expects that there is a car that hit the pole and classifies it as an accident. Further research is needed to address these limitations and improve the accuracy and effectiveness of the system. as we can see in the next figure

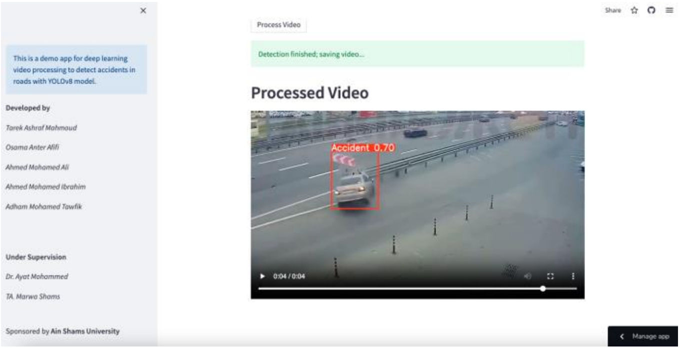

Figure 9 : Process Video Example "Accident Detection"

Overall, our project demonstrates the potential of using Computer Vision techniques for car accidents detection. By using a custom dataset and training a Deep Learning model, we were able to develop a system that can detect car accidents in real-time with a high degree of accuracy. The system has a significant implication for improving road safety.

Conclusion and Future Work

The car accident detection system is a crucial technology that can help reduce the number of road accidents and save lives. In this project, we designed a website that takes videos as input and detects accidents in any frame. We trained the YOLO v8 model to detect accidents and collected a diverse dataset to improve its accuracy. Our system architecture includes several components that work together seamlessly to provide a reliable and efficient system that can detect accidents and alert emergency services in real-time. The advantages of our project include its ability to detect accidents quickly and accurately, which can help reduce the response time of emergency services. However, there are also some disadvantages such as the need for a stable internet connection to access the website and the potential for false positives or negatives. Further testing and validation can help address these issues and improve the overall reliability of the system.

There are several open points in our project that can be continued in the next year. Improving the accuracy of the accident detection module, adding more features to the alert system module, enhancing the user interface of the website, and integrating the system with other technologies can contribute to the creation of a safer and more sustainable transportation system. With further development and refinement, the system can provide even more reliable and efficient accident detection capabilities.

The potential impact and benefits of a car accident detection system are significant, and further development and implementation of the technology can contribute to the creation of a safer and more sustainable transportation system. The system can help reduce the number of fatalities and injuries caused by road accidents, improve the overall efficiency and safety of transportation systems, and benefit a wide range of users.

Acknowledgment

We would like to express our gratitude to several individuals and organizations who have contributed to the success of this project. Firstly, we would like to thank TA Marwa Shams for his valuable guidance and support throughout the project, We cannot forget the role of **Dr. Mona Abdelazim** for her efforts, We are grateful to **Dell Technologies** for providing us with the necessary resources to carry out this research, Specially by Eng. Omar Nael, Finally, we would like to thank the **Ministry of Communications and Information Technology (MCIT)** for their support and encouragement of our research.

References

[1] https://github.com/Cogito2012/CarCrashDataset

[2] https://www.youtube.com/watch?v=KrJTpYsWe3w

[3] 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. [Google Scholar] [CrossRef]

[4] https://www.youtube.com/watch?v=wuZtUMEiKWY&start=774

[5] https://anvil.works/learn/tutorials/google-colab-to-web-app

[6] SALIL SRIVASTAVA, PRIYANSHU KUMAR ,JAGMOHAN DAS BAIRAGI ;TUSHAR AGARWAL ;GAURAV MEHROTRA ; Car Crash Detection using YOLOv3. May 18, 2020. [Google Scholar]

[7] https://www.youtube.com/watch?v=gKTYMfwPo4M

[8] https://ijcsmc.com/docs/papers/May2022/V11I5202216.pdf

[9] Nancy, P.; Dilli Rao, D.; Babuaravind, G.; Bhanushree, S. Highway Accident Detection and Notification Usingmachine Learning. Int. J. Comput. Sci. Mob. Comput. 2020, 9, 168–176. [Google Scholar]

[10] Ghosh, S.; Sunny, S.J.; Roney, R. Accident Detection Using Convolutional Neural Networks. In Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 1–2 March 2019. [Google Scholar] [CrossRef]

[11] Ijjina, E.P.; Chand, D.; Gupta, S.; Goutham, K. Computer Vision-based Accident Detection in Traffic Surveillance. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019. [Google Scholar] [CrossRef][Green Version]

[12] Gollapalli, M.; Rahman, A.U.; Musleh, D.; Ibrahim, N.; Khan, M.A.; Abbas, S.; Atta, A.; Farooqui, M.; Iqbal, T.; Ahmed, M.S.; et al. A Neuro-Fuzzy Approach to Road Traffic Congestion Prediction. Comput. Mater. Contin. 2022, 73, 295–310. [Google Scholar] [CrossRef]

[13] Ahmed, M.I.B.; Rahman, A.-U.; Farooqui, M.; Alamoudi, F.; Baageel, R.; Alqarni, A. Early Identification of COVID-19 Using Dynamic Fuzzy Rule Based System. Math. Model. Eng. Probl. 2021, 8, 805–812. [Google Scholar] [CrossRef]

[14] Rahman, A.; Basheer, M.I. Virtual Clinic: A CDSS Assisted Telemedicine Framework. In Telemedicine Technologies; Academic Press: Cambridge, MA, USA, 2019; pp. 227–238. [Google Scholar]

[15] Alotaibi, S.M.; Rahman, A.U.; Basheer, M.I.; Khan, M.A. Ensemble Machine Learning Based Identification of Pediatric Epilepsy. Comput. Mater. Contin. 2021, 68, 149–165. [Google Scholar] [CrossRef]

[16] Rahman, A.-U.; Abbas, S.; Gollapalli, M.; Ahmed, R.; Aftab, S.; Ahmad, M.; Khan, M.A.; Mosavi, A. Rainfall Prediction System Using Machine Learning Fusion for Smart Cities. Sensors 2022, 22, 3504. [Google Scholar] [CrossRef] [PubMed]

[17] https://www.youtube.com/watch?v=PSpO9PM8NS4