DETECTION OF ROAD TRAFFIC CRASHES BASED ON COLLISION ESTIMATION

Mohamed Essam, Nagia M. Ghanem and Mohamed A. Ismail

Department of Computer Engineering,
Alexandria University, Alexandria, Egypt

ABSTRACT

This paper introduces a framework based on computer vision that can detect road traffic crashes (RCTs) by using the installed surveillance/CCTV camera and report them to the emergency in real-time with the exact location and time of occurrence of the accident. The framework is built of five modules. We start with the detection of vehicles by using YOLO architecture; The second module is the tracking of vehicles using MOSSE tracker, Then the third module is a new approach to detect accidents based on collision estimation. Then the fourth module for each vehicle, we detect if there is a car accident or not based on the violent flow descriptor (ViF) followed by an SVM classifier for crash prediction. Finally, in the last stage, if there is a car accident, the system will send a notification to the emergency by using a GPS module that provides us with the location, time, and date of the accident to be sent to the emergency with the help of the GSM module. The main objective is to achieve higher accuracy with fewer false alarms and to implement a simple system based on pipelining technique.

KEYWORDS

RTCs, ViF, SVM, Deep Learning, Collision Estimation.

1. INTRODUCTION

Nowadays, the usage of vehicles increases with the corresponding rise in population. Consequently, accidents are increasing as well due to different reasons. The world health organization (WHO) states that RTCs are in the top 10 reasons that cause death; there are more than 1.35 people die and 50 million injuries each year because of RTCs [1][2]. In addition, in Egypt, there are nearly 12000 Egyptians die, and thousands of people injure because of RTCs. RTCs will increase continuously to become the top cause of death by 2030 [3]. The main causes of accidents are due to over speed, driver inattention, blown tire, and wrong passing as shown in Figure 1. Also, the delay in reporting the accident and the delay in reaching the ambulance to the accident location are considered to be one of the main reasons [4]. In addition, RTCs that happen in remote places are very difficult to be traced. So, it’s a challenge for the emergency services to get to the exact location of the accident which results in death. Fortunately, governments in developed and upper-middle-income countries installed a large number of CCTV cameras on roads. For example, in China, 200 million CCTV cameras have been installed [5]. In London city, there are 500 thousand CCTV cameras [6] and with the availability of the huge processing power of computers nowadays, such as cloud computing services and relatively affordable hardware. This paper aims to detect RTCs depending on CCTV cameras installed, report to the emergency services in real-time in order to recover victims, and allow them to monitor the accident using a client-server architecture and an interactive GUI.

David C. Wyld et al. (Eds): ICAIT, CBIoT, WiMo, CRYPIS, ICDIPV, CAIML, NLCA - 2022
pp. 169-179, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121213

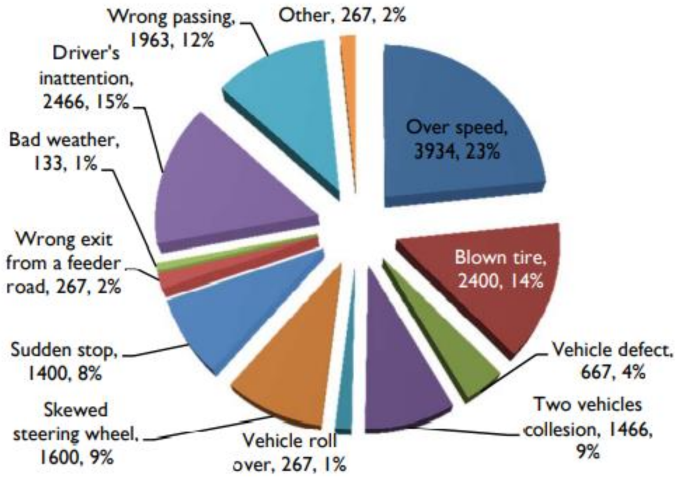


Figure 1. Number & Percentage of main causes of crashes in Egypt

2. RELATED WORK

Many authors have worked on the topic of RTCs. Some of them worked by using deep learning like this one [7]. However, the deep learning method is not the best solution because of the shortage of video crash datasets. In [7], they used a very low number of videos which makes it difficult to solve a complex problem like RTCs detection. Moreover, the accident datasets need to cover different types, different scenarios, and different ways that need to be fed to the neural networks, the network can be considered a good solution. There are also some papers that depend on feature extraction like this paper [8]. The researchers describe three stages of systems to detect RTCs, starting with the first stage which detects the car then the damaged texture detector with SVM which recognizes damaged parts, and finally, the car parts detector which detects the car's parts, the accuracy is about 81.83%. Also, in [9], the authors discussed single-vehicle traffic accident detection which consists of an automated traffic region detection method, a traffic direction estimation method, and a first-order logic traffic accident detection method, their proposed methods achieve good performance in real-time RTCs. Also, in [10], the authors proposed a three stages framework to detect RTCs. They start with a car detection method and then a tracker to focus on each car then in the final stage, they used the ViF descriptor that was introduced in this paper [11]. They got 89% accuracy. Their approach succeeded to be a general solution as it achieved high true alarms. But unfortunately, they got a lot of false alarms which made the system unreliable to detect the accidents. The bad accuracy was because of a lack of dataset of accidents and the accidents are stochastic.

The last approach was the most promising one, and we continued on their work with a method to find a way to maintain their high true alarms while achieving very low false alarms, and that is why we introduced our system for detecting accidents based on collision estimation.

After introducing the previous approaches to solving the problem, we found that they depend more on the clarity of the CCTV cameras and assuming that all CCTV cameras have exact quality resolution and are installed at a specific height and capture the vehicles at the same scale, which is not in real life.

Our approach seeks to work well on most CCTV cameras on roads. Moreover, they depend on the availability of a dataset of road accidents in the future which will increase the accuracy of their approaches, but what we propose here does not depend on the dataset as we solved the problem as if it was a 2D car game in a 3D world and dealing with the problems of different resolution and different scales as we will explain later.

3. Proposal

The framework consists of 5 modules; it starts with the vehicle detection module using YOLO neural network. In the second module, we track each vehicle using the MOSSES tracker. A third module is a new approach to detecting accidents based on collision estimation. Then the fourth module for each vehicle, we detect if there is an accident or not by using ViF descriptor with an SVM classifier to detect vehicle crashes. Finally, if there is an accident, it will report the emergency using GPS and GSM modules. Figure 2 illustrates the flow of the system.

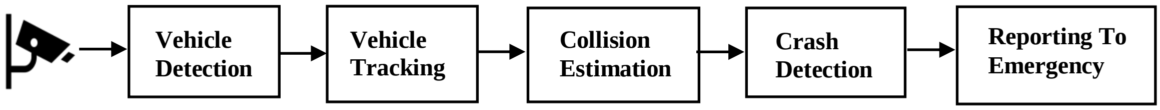
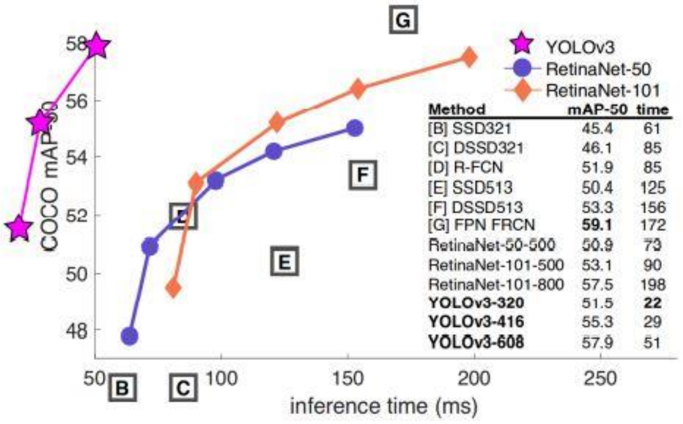


Figure 2. Proposed framework for detecting accidents

3.1. Vehicle Detection

For vehicle detection, we have used You Only Look Once (YOLO) network proposed in [12] because of its high accuracy and very low time processing compared to different networks as shown in Figure 3. In this paper, we have used YOLOv3 [13]. Yolo is considered faster than other convolutional networks because it looks at the image once and derives the bounding box and class probability from each object. A grid of SxS boxes has been used to divide the image. Then compute the confidence score for each box to see if the bounding box contains an object or not. So, the higher the confidence score gets the higher probability that the bounding box contains an object. When there is a single object surrounded by multiple bounding boxes. non-maximum suppression is applied to keep the most robust detection around a single object.

So, the results of this stage are the bounding box and labels that passed to the tracking module where the tracking module uses these bounding boxes as a starting point to track the vehicles.



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Figure 3. Comparison between the performance of YOLOv3 and other networks [13].

3.2. Vehicle Tracking

There are many tracking algorithms. But we need the type of tracking algorithm that fits our problem, that acts well on rotations, occlusion, and other distractions, that depends on correlation filters, so we used an algorithm called Minimum Output Sum of Squared (MOSSE) [14]. MOSSE depends on the frequency domain. It is capable of quickly obtaining the frequency transformation of the vehicle image using fast Fourier transform. With each new frame, this correlation filter is updated online. MOSSE filter is a correlation filter and that's why it can track complex objects. In addition, it produces a stable and reliable correlation filter when initialized using a single frame. It is also strong to variations in illumination, scale, position, and non-rigid deformations as well.

We are not only focusing on increasing the accuracy but also focus on performance. Vehicles are tracked every frame for 30 frames. And due to the congested road or traffic lights, some vehicles may not move or move at a slow speed. This is causing overprocessing on the tracking module to track the vehicle in every frame. So, we propose track compensated frame interpolation (TCFI) by estimating the vehicle's speed. If the vehicle's speed in the last three frames was less than the minimum speed, rather than tracking it in every frame, we track it in one frame and estimate where it will be in the following frame without using the tracking technique. So, the TCFI algorithm should be reevaluated in every frame since the vehicle may move higher than the minimum speed in some frames. Hence, in this situation, we will move it to the tracking algorithm then.

Applying TCFI improves the framework performance a lot, especially on congested roads, as the tracking module will use approximately half of the performance and will have the same accuracy as before.

3.3. Collision Estimation

In the beginning, we have an initial set of detected vehicles, then we create a unique id for each detected vehicle, and then we track each of these vehicles and maintain their id as they move through frames in a video. Now we have an object for each vehicle. The vehicle's position is saved for 30 frames in that object. The idea of the new approach is to limit the trackers entering the ViF descriptor using various features.

The collision estimation module might be used as a classifier to identify whether a crash happened or not, using the following algorithms.

First, we need to estimate the vehicle's speed in the video. Having a tracker on every vehicle, we can easily estimate the average speed of the vehicle by pixel unit, However, the camera angle differs from one CCTV camera to another as well as the camera position, height, and resolution. So, we have to keep in mind that we are dealing with a variety of CCTV cameras on the road. Thus, another unit instead of a pixel unit is needed to estimate the average speed of a tracker. To solve the camera resolution problem, every input feed must be resized to fixed width and height (480,360). However, if the majority of the CCTV cameras have a resolution of (1920, 1080), it is better to resize the input feed to that resolution. Another problem arises when CCTV cameras are hung at various heights or the feed is captured on various scales. As the area of the vehicle varies in the video, the vehicle will appear small if the camera is placed at a distant height. However, if the camera is set at a low height, it will appear that the vehicle has a large area; and because we estimate speed using pixel units, the small area will appear that moves slower than the larger area. So, the height or scale problem can be solved by multiplying the average speed of a vehicle by a speed coefficient parameter stated in Equation 2. This parameter has an inverse relation to the area; a larger vehicle has a lower speed. But we also take into consideration the different types of vehicles as the motorcycle area will appear less than the truck area. This is solved by α parameter that is set according to each type of vehicle. For instance, a car has (α = 4). We calculate the coefficient as shown in Equation 1. sum_dx in Equation 2 represents the vehicle's horizontal movement during the last 10 frames.

$$\text{Coefficient} = \frac{Area\ of\ video\ frame}{\alpha \times \text{Area}\ of\ vehicle} \tag{1}$$

$$Speed = \text{Coefficient} * \frac{\sqrt{sum\_dx^2 + sum\_dy^2}}{10} \tag{2}$$

The angle of the camera plays an important role in the collision estimation process. The best camera angle is the one with the top view since the focus is on the x and y plane where the vehicles move. In the real world, it is impossible to capture such a top view image unless you use a drone as a CCTV camera. this makes the perfect possible perfect CCTV angle is the one in Figure 4 instead of Figure 5. If you can't modify the camera angle, the vehicle's average speed estimation and future position will suffer, reducing the system's accuracy; however, this may be fixed later in the next module.

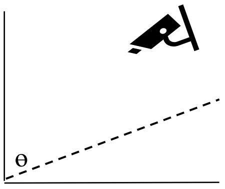


Figure 4: CCTV camera capture the view with an angle 30° to 90°

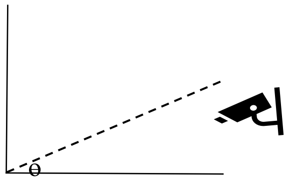


Figure 5: CCTV camera capture the view with an angle 0° to 30°

Second, compare every two vehicles with each other in different frames. If the average speed of the two vehicles is less than the speed limit, which is a hyper-parameter, then the two vehicles will not collide. As they are both moving slowly, So, we can reduce the possibility of their colliding at this stage. However, if one or both of them are moving faster than the speed limit, there is a risk of a collision, and the next step should be taken. This process of discarding some results is considered an excellent optimization of the computational power, especially when dealing with congested roads. So, only the high-speed cases will be investigated.

Third, measure the estimated centers in future frames for both vehicles after 10 frames of their current frame by calculating the average speed of the last 10 frames, as well as their angle, and predict the vehicle's position after ten frames from the current frame.

Fourth, measure the distance between the two estimated centers vehicles for both vehicles. We do that as we need to limit the cases where the vehicles will not meet close to each other in the future. So, if the distance between the two estimated centers is greater than half of the distance between the vehicle's center and its corner, plus the distance between the other vehicle's center and its corner, then they will not collide because they are apart from each other. But if the distance is below, then the next step should be taken. So, only high speed and close to each other cases will be investigated.

At this stage, high-speed vehicles and close range to each other will report an accident but if the camera angle is not perpendicular to the road as shown in Figure 5 which is what most CCTV cameras are. Then an occlusion will occur and it will show as if they crash into each other. To solve this, we will calculate the difference between the estimated future and actual position for every two vehicles and get the maximum distance. If the maximum distance between two vehicles' actual and estimated centers is greater than half the distance between their two estimated centers in the future, it may be a crash, but if it is less, it is not a crash and they are limited out.

The results of detecting RTCs using collision estimation only without the next module crash detection were excellent. However, we also tried to add the next module crash detection to the pipeline to evaluate the performance of each module and determine the best technique to detect RTCs.

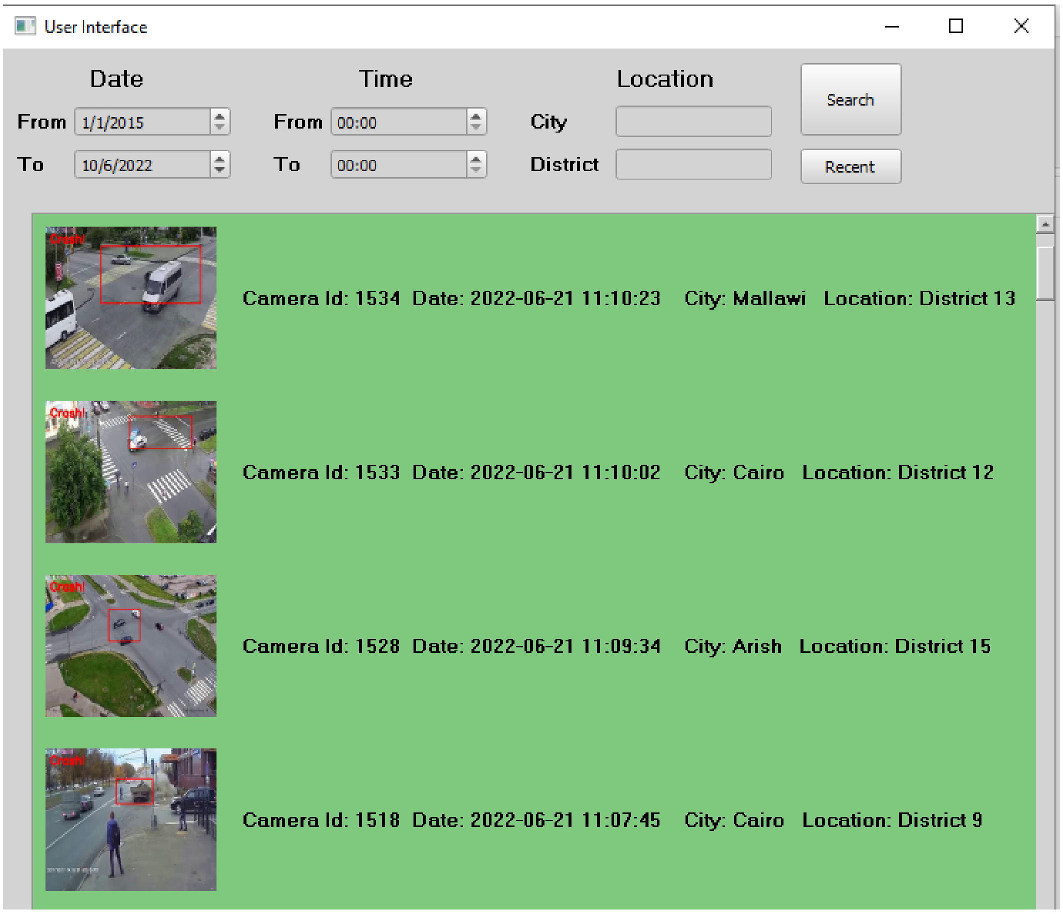
3.4. Crash Detection

We added another module to increase accuracy by tackling the camera angle problem mentioned in the collision estimating module after filtering out bad candidates and categorizing accidents. As a result, a feature vector is obtained, which is then used as input to a support vector machine (SVM) model that identifies whether there has been an accident or not. The feature vector for a single vehicle's sequence of frames is obtained using the violent flow (ViF) descriptor [11]. As an optical flow algorithm, we used ViF descriptor and Horn-Schunck. We used it for its good accuracy and minimal computing cost, as explained in this [10]. ViF descriptor is based on statistics of change in the magnitude of the optical flow vectors.
Finally, an SVM model is trained on the obtained feature vector from the ViF descriptor and classifies if there is an accident or not.

3.5. Reporting to the Emergency

In the last module in our pipeline, if the system detected an accident, it will not only send an SMS to the emergency services with the help of the microcontroller, GPS, and GSM modules but it will also send a notification in nearly real-time and allow them to monitor accidents using client-server architecture and an interactive GUI. In addition, the system saves accidents in the database indicating the location, date, time, and video of the accident where they can inspect it at any time as shown in Figure 6.

In Figure 7, we show the architecture of our system where it begins with the backend which collects the frames from the CCTV camera and sends them to the model passing through the four modules to determine whether there has been an accident or not. And once an accident happens, it will send an SMS to the emergency services and a notification to display the accident to check the criticality of the accident and take an urgent action. This helps to save many lives by reporting them in time.



User Interface

Date		Time		Location		
From	1/1/2015	From	00:00	City		Search
To	10/6/2022	To	00:00	District		Recent

Camera Id: 1534 Date: 2022-06-21 11:10:23 City: Mallawi Location: District 13

Camera Id: 1533 Date: 2022-06-21 11:10:02 City: Cairo Location: District 12

Camera Id: 1528 Date: 2022-06-21 11:09:34 City: Arish Location: District 15

Camera Id: 1518 Date: 2022-06-21 11:07:45 City: Cairo Location: District 9

Figure 6. The system saves the accidents indicating date, city and location.

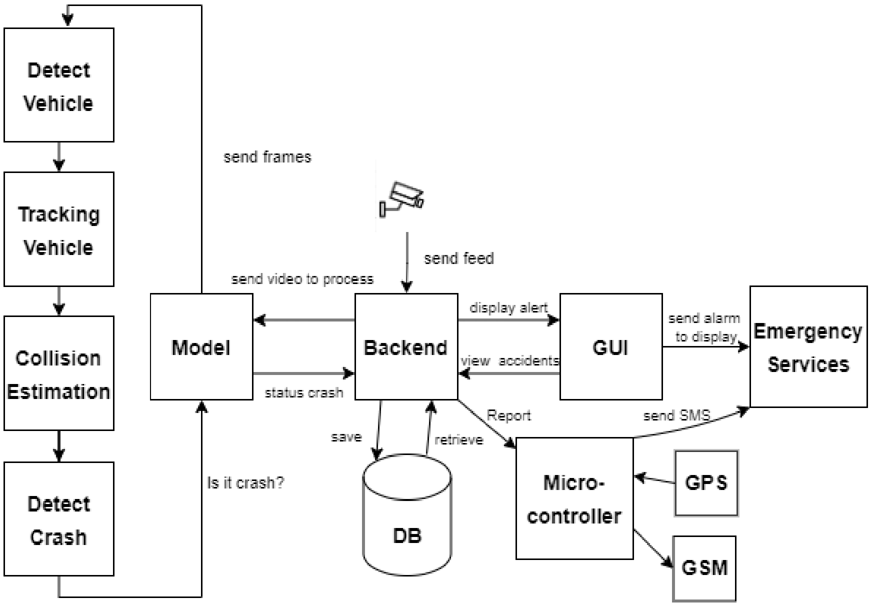


Figure 7. Architecture of the system

4. RESULTS

The experiments were done on a computer running Windows 10, Intel Core i7-10510U CPU @ 2.30 GHz with 16 GB of RAM and NVIDIA GeForce 920MX, and the system runs on python 3.6.

4.1. Datasets

We had to collect two types of the dataset; the first for training the ViF descriptor, which will be as described in 30 frames per second for the crashed vehicle only, and the second dataset for testing the whole system and the length of its video are about 20 seconds to 120 seconds.

First, We collected the training dataset for the ViF descriptor in three steps, the first was searching for an available dataset till we found those two papers [10][15], filtering them to obtain a high-quality resolution for vehicle crashes and the last step was executing our system and getting the output, which we then fed the training dataset with it. So, in total after those steps, we got 200 videos.

Second, we gathered the test datasets by downloading them from different YouTube sources, cutting and editing them, and getting 75 videos to test the system.

4.2. Results

We proposed two approaches, one depends only on collision estimation, and the other depends on both collision estimation and ViF descriptor. The system which depends only on collision estimation gives higher recall but lower accuracy. While the system which depends on both collision estimation and ViF descriptor have a lower recall but better accuracy as shown in Table 1.

In Figure 8; shows a comparison in processing performance. Our system is better in performance, especially on congested roads.

In Figure 9, we show our results in detection crashes using collision estimation. we mark the exact position of car crashes with red boxes.

Also, we measure the processing time of our system from the vehicle detection stage till emergency notification stage to get 3.04 seconds only for processing a video of 5 seconds as shown in Table 2.

Table 1. Accuracy and recall of our system compared to others

Method	**Accuracy**	**Recall**
Deep Spatio-Temporal Model [7]	79%	77%
ViF Descriptor [10]	75%	80%
Collision Estimation	91%	**94%**
Collision Estimation + ViF Descriptor	**93%**	78%

Using two techniques, our system performs better, especially on congested roadways. First, by applying TCFI to our tracking algorithm where the tracking algorithm applies only to the vehicles that move fast and apply TCFI to vehicles that move slowly or stopped. Second, we managed to limit the number of trackers entering the ViF descriptor using collision estimation which results in very fast and high performance compared to using only ViF.

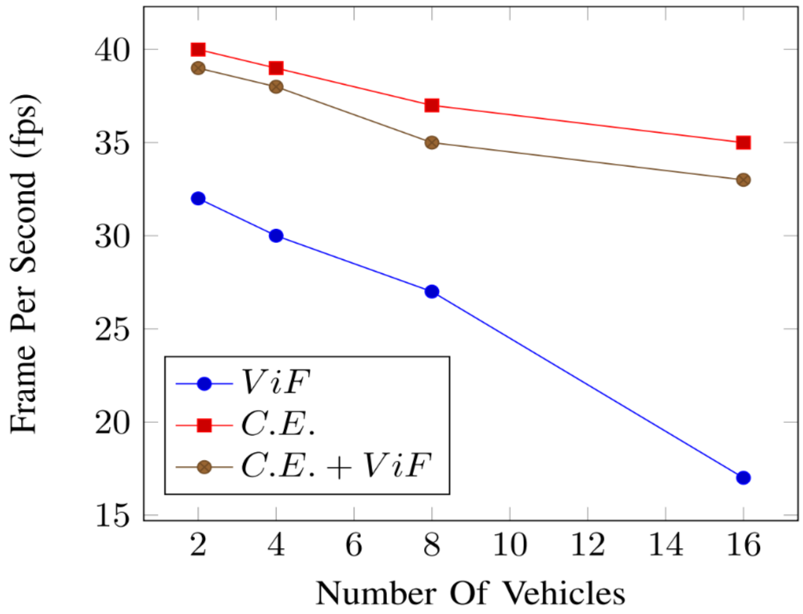


Figure 8. Comparison of our system and others in performance of frames per second with various numbers of vehicles

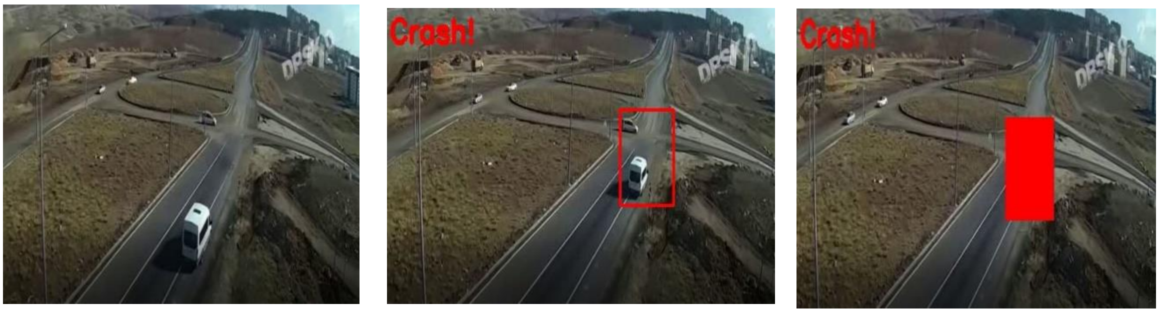


Figure 9. Car crash detection using collision estimation

Table 2. Processing Time of our system using collision estimation

	Video duration in sec.	Time processing in sec.
Collision Estimation	5	3.04

5. Conclusion

We presented a system that detects RTCs using installed CCTV cameras in real-time based on collision estimation. We also proposed a new technique track-compensated frame interpolation (TCFI) to track vehicles in a more efficient manner, especially on congested roads. The system achieves excellent results with low processing and performed better than other systems, with a 94% recall rate and 93% accuracy rate.

REFERENCES

[1] "The top 10 causes of death", WHO, Dec 2020, [online] Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] "Road traffic injuries", WHO, June 2021, [Online] Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[3] "Egypt Road Safety", WHO, 2012, [online] Available: http://www.emro.who.int/images/stories/cah/fact_sheet/road_safety.pdf.

[4] Vaishnavi Ravindran, Lavanya Viswanathan, and Shanta Rangaswamy, "A Novel Approach to Automatic Road Accident Detection using Machine Vision Techniques", International Journal of Advanced Computer Science and Applications, vol. 7, Nov 2016.

[5] Coco Feng. "China the most surveilled nation? The US has the largest number of CCTV cameras per capital". In: (Dec. 2019). URL: https : / / www . scmp . com / tech / gear / article / 3040974 / china - most - surveilled - nation - us - has - largest - number-cctv-cameras-capita.

[6] Jonathan Ratcliffe. How Many CCTV Cameras in London? URL: https://www.cctv.co.uk/how-many-cctv-cameras-are-there-in-london/

[7] Singh and C. K. Mohan. "Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder". In: IEEE Transactions on Intelligent Transportation Systems 20.3 (2019), pp. 879–887. DOI: 10.1109/TITS.2018. 2835308.

[8] Ravindran, L. Viswanathan, and S. Rangaswamy, "A novel approach to automatic road-accident detection using machine vision techniques," INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, vol. 7, no. 11, pp. 235–242, 2016.

[9] Ghahremannezhad, Hadi & Shi, Hang & Liu, Chengjun, "A Real-Time Accident Detection Framework for Traffic Video Analysis". In 16th International Conference on Machine Learning and Data Mining(2020).

[10] V. Machaca Arceda and E. Laura Riveros. "Fast car Crash Detection in Video". In: 2018 XLIV Latin American Computer Conference (CLEI). 2018, pp. 632–637. DOI: 10.1109/CLEI. 2018.00081.

[11] Vicente Machaca, J.C. errez, and K. n. "Real Time Violence Detection in Video". In: Jan. 2016, 6 (7.) –6 (7.) DOI: 10. 1049/ic.2016.0030.

[12] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[13] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: CoRR abs/1804.02767 (2018). arXiv: 1804. 02767. URL: http://arxiv.org/abs/1804.02767.

[14] David S. Bolme et al. "Visual object tracking using adaptive correlation filters". In: June 2010, pp. 2544–2550. DOI: 10. 1109/CVPR.2010.5539960.

[15] A. P. Shah et al. "CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis". In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). 2018, pp. 1–9. DOI: 10.1109/AVSS.2018. 863

AUTHORS

Mohamed Essam Ahmed
Teaching Assistant of computer science, Alexandria University,
Faculty of Engineering.

Nagia M. Ghanem
Associate professor of Computer Science, Alexandria University,
Faculty of Engineering.

Mohamed A. Ismail
Professor of Computer Science, Alexandria University,
Faculty of Engineering.