Vision-based automatic warning system to prevent dangerous and illegal vehicle overtaking

Mahinsa Athree*
Department of Statistics and Computer Science
University of Kelaniya, Faculty of Science, Sri Lanka
mahinsaathree@gmail.com

Anusha Jayasiri
Information Technology Centre
University of the Visual and Performing Arts, Sri Lanka
anusha.j@vpa.ac.lk

Abstract: **The purpose of this research is to implement a system that can be a help for drivers to drive vehicles safely and reducing accidents. Most of the accidents have occurred when overtaking. The driver has to consider the distance between two vehicles and the speed of the front vehicle. If there is another vehicle moving the same or opposite direction, it must be considered as well. Also, traffic signs, solid line, and a broken line must be considered. Reducing the errors which can be happened by the driver when deciding all these stuff in instinct is the goal of the "Automated Vehicle Overtaking System". This research has been used three pipelines, such as lane line detection, traffic sign detection, and vehicle detection. The input video stream from the front view camera of the vehicle is transferred through the above pipelines. The system gives the final output frame with confirmation that 'is it safe to overtake now'. The driver can recognize the danger by alarm sound from the system. This system has a Python Application Programming Interface (API). Using that API, any device or application can access this system. Finally, the main goal of this research is to implement a system that can predict dangers in overtaking and saves the lives of passengers and vehicles.**

Keywords: Flask, Image Processing, Object detection, OpenCV, Python API

I. INTRODUCTION

As the number of vehicles on the road increases exponentially, road accidents are also increasing proportionately. Although road rules, traffic signs, etc. play an important role in ensuring the safety of vehicles on the road, recent statistics show the number of accidents on the road is unacceptably high.

Different types of traffic signs are used for better traffic management. Danger warning signs, regulatory signs, directional informative signs, traffic light signals, and road markings are the categories of traffic signs used in Sri Lanka. Besides, Sri Lankan roads are marked with solid yellow lines, solid white lines, solid double yellow lines, solid double white lines, broken white lines and broken yellow lines.

Vehicle manufacturers are also adding more and more safety features to their products to make them safer as well as a way of competing with each other. As a result of these efforts, modern vehicles are much safer than previous generations. Even with these technologies still, numerous accidents take place. Many accidents occur when drivers attempt to overtake vehicles.

The purpose of this research is to warn drivers when overtaking through a camera-based system. Front view and rearview cameras are used in vehicles can introduce an additional layer of safety. Front view cameras, generally known as dashcams are very useful to see a full view through the windscreen. Detected frames are shown on a display of the vehicle or drivers can view the video stream through a mobile device. Some dashcams can connect other devices via Wi-Fi, Bluetooth, and 4G.

It is better for drivers if there is no traffic, no pedestrians, no traffic lights and straight lanes, but it is not the reality. The driver has to take care of all of the above-mentioned stuff. If there is a pedestrian crossing traffic sign, the driver has to drive carefully. Because there can be pedestrians on the road. If there is a traffic light sign, the driver has to follow those signal lights. The driver has to pay attention to traffic signs, road markings, vehicles in front of his vehicle and vehicles behind as well.

When overtaking, the proposed system warns drives based on the speed of the vehicle, speed of the vehicle in front, the distance between two vehicles, road marking and traffic signs.

II. RELATED WORK

Hadi et al reviewed image processing tools that can be used for implementing vehicle detection and tracking systems. Also, this review contains what analysis tool is more effective when implementing the above-mentioned applications. Most effective preprocessing techniques can be identified by this review [1].

Ali et al introduced the "Safe passing System of Vehicle" system that facilitates to overtake safely with a microcontroller circuit. For this purpose, a measuring system sensing element was employed. The measuring system sensing element detects the gap and existence of a car. The signal is regenerated into visual through different colour light-emitting diode lightweight by a microcontroller circuit [2]. Image processing or computer vision techniques for detecting speeds of vehicles have not been used in this work.

Gholami et al proposed a new algorithm that uses the digital video, image processing and computer vision to automatically detect vehicle speed. The captured traffic videos are collected with a stationary camera that is mounted on a freeway. Since the foreground of an image represents the moving objects only, the speed of a vehicle is calculated by removing the background and extracted the foreground of the image sequence. The speed of the vehicle in each frame is calculated using the position of the vehicle in each frame. [3]. Redmon et al had implemented the Yolov3 that has a new classifier network better than others. Bounding boxes are predicted utilizing dimension clusters as anchor boxes by YOLO9000. This network calculates four coordinates for each bounding box such as x, y, w, h. According to the paper x and y are coordinates of the centroid of the bounding box. Also, w

and h are the width and height of the bounding box [4]. The proposed system prefers to use the Yolov3 network because its accuracy is 51.5 mean Average Precision (mAP) and speed is 45 frames per second (fps). Those values are better than other networks' accuracy and frame speed.

Bhaskar et al introduced an automatic vehicle investigation system using videos recorded from stationary cameras close to roads. Primarily image processing and object detection were used for implementing this model, furthermore, the Gaussian mixture model and blob detection methods were used for implementing this algorithm. Morphological techniques were used to remove noise from input frames. Final calculations were done by detected objects and regions [5].

Rosebrock calculated the distance between the camera and the detected object using Triangle similarity. Initially, an object was kept at a known distance from the camera and then taking a photo of it.

$$F = (P \times D) / W \qquad (1)$$

The focal length of the camera can be calculated using (1). P, D, and W are apparent width in the image, known distance and actual width respectively.

$$D = (W \times F) / P \qquad (2)$$

When the focal length of the camera is known, distance calculation can be determined by (2). As mentioned earlier, W is the actual width of the object. F is the calculated focal length. P is the apparent width of the object in pixels. Finally, the actual distance is calculated [6].

Low et al presented a system that identifies a road as well as the left and right boundaries of the road. A vision sensor was utilized to capture the front view frames of the vehicle. Edge detection algorithms and Hough transform algorithms were used to process the frame sequence. Using Hough transformation, the boundaries of the road can be drawn according to the edges that were recognized by Edge detection [7].

III. METHODOLOGY

This research consists of four main stages. Every single input frame passes these steps to extract important information and use those data to have the final result. The approach is shown in Fig. 1.

A. Road marking detection

As the first step, a pipeline has been implemented to detect lines on the road such as a solid line and a broken line in both yellow and white colours. The "Road marking detection" pipeline starts with input frames from the front view camera of the vehicle and ends with the final road marking detected frames.

The method of "Road marking detection" used several image processing techniques such as Grayscale, Gaussian Blur, Canny edge detection, Region selection, Hough line transformation and weighed image methods.
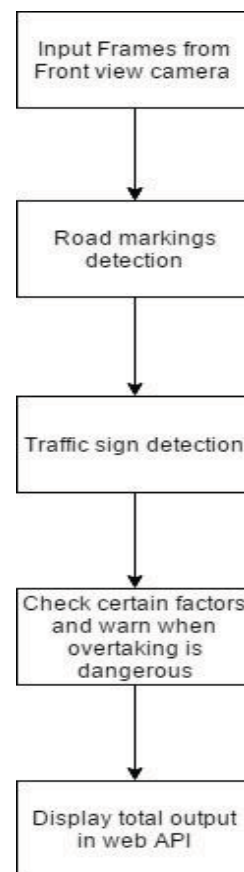

Fig. 1. The Overall approach of the research

The grayscale method is used to convert a multichannel image into a single-channel image. Because of the blurring, an image reduces the noise and unnecessary details of an image, Gaussian Blur method is used for the blurring of the Grayscale image. The Canny Edge detector is used to detect the edges of the images which is to remove the unnecessary information and extracted only the important data of an object [8]. The following Fig. 2 shows the edge detected and original images.
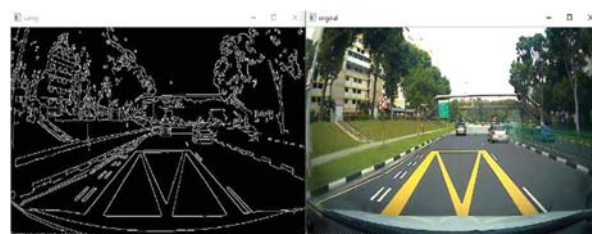

Fig. 2. Canny edge detected image and the original image

The region selection method was used to extract important and relevant regions of the frame concerning the road. Once the unwanted regions of the frame have been removed, the processing power can be reduced. Fig.3 shows an example of a region selection of a road. The size of the region was selected by changing the values of the bottom left, top left, bottom right, and top right vertices.
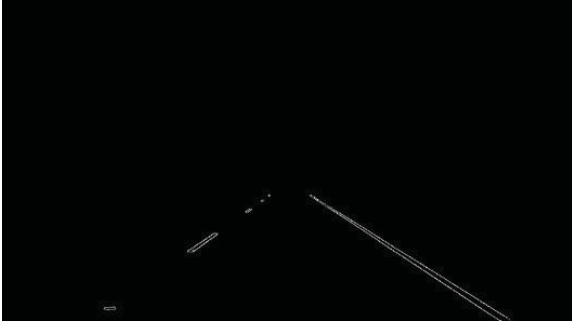
26

Fig. 3. Lane region of the image after region selection

Hough line transformation method was used to detect line segments in the frame. "HoughLinesP" is the method that was used in this system to detect lines in frames which is a default method in OpenCV. The system used the two attributes minimum line length and maximum line gap for detecting solid lines on the road but not for detecting broken lines. For better performance of the system the "Min line length" value is increased and the "Max line gap" value is decreased. AddWeighted function in OpenCV by default also used to add weighted features into the original frame. After applying all the above-mentioned methods, a detected road marking of a road is shown in Fig.4.

Fig. 4. Detected lanes in a road

B. *Traffic sign detection*

The detected road markings image is the input for this stage of the process and a traffic sign detected image is the output. Several subsidiary steps are involved in the process. The system must have Haar cascade files for each traffic sign that required to detect. The methodology for implementing the cascade file is presented in Fig.5.

To train the Haar cascade files for traffic signs, positive images and negative images were used. The images which are needed to be detected in the positive image list and any images other than the subject image are in the negative images list. In this research, 65 positive images and 1975 negative images for each traffic sign were used to train the system.

When a positive image set was created, there were different sizes of images. Therefore, all the images were resized into 50*50 size. As negative images contained images of various sizes and useless images, those useless images were removed and selected images were resized to 100*100 size. A vector file was created using the list of positive image samples with some additional information for training tasks.

As the next step, by giving negative and positive data, the classifier was trained using the OpenCV function Traincascade utility.
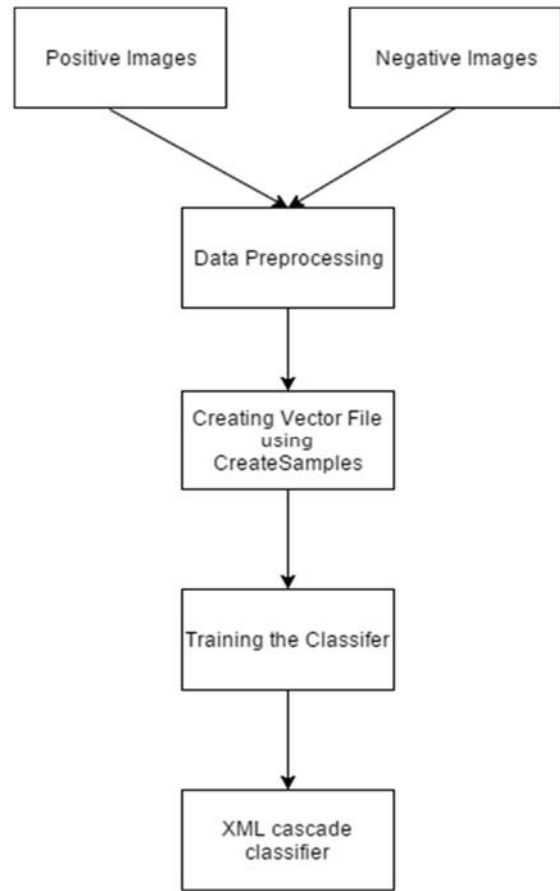

Fig. 5. Training of Haar Classifier

After generating the cascade Extensible Markup Language (XML) file successfully, it could be used to detect the traffic sign that was targeted when training the cascade file. There could be true detection and false detection. True detection needs to be higher than false detection to have efficient detections.

$$\text{True Detection} = \frac{\sum_{i=1}^{N} h_i}{\sum_{i=1}^{N} c_i} \qquad (3)$$

$$\text{False Detection} = \frac{\sum_{i=1}^{N} f_i}{N} \qquad (4)$$

True detection and False detection were calculated from the two equations (3) and (4) given above. In (3), c_i is the number of call numbers in test images and h_i is the number of positive hit occasions. In (4), N is the number of test images and f_i is the false rate of each test image. A hit rate is several true detected call numbers in an image and a false rate is several false detected call numbers [9]. An example of a traffic sign detected image is shown below in Fig.6.

27

Fig. 6. Detected traffic sign

C. *Vehicle detection*

Vehicle detection is used to detect vehicles that are travelling in front of the subject vehicle. These detected vehicles can be either incoming vehicles or frontier vehicles that travel the same way as the subject vehicle heading. The algorithm is capable of not only detecting vehicles but also calculating distances and speeds of the vehicles relative to the subject vehicle.

Obtaining a blob from the image, loading the blob image to network, drawing bounding box and distance calculations are the main functions of the vehicle detection process.

Image frames were preprocessed to make correct predictions from the deep neural network. Mean subtraction, scaling and optionally channel swapping were performed. When implementing the network requires a weight file, a configuration file, and a text file. This text file contains all the classes that the system needs to detect.

You Only Look Once (YOLO) algorithm was used for the real-time object detection because this approach is faster and accurate. The coco data set was used in pre-trained weight files.

Drawing a bounding box was the most important part of this vehicle detection set up. The width of the bounding box was used when calculating the distance between two vehicles. Bounding boxes were represented by the four items (x, y, w, h), where x, y are the top left corner, and w, h are the width and height respectively. The detected vehicle image is shown below in Fig.7.

Fig. 7. Vehicle detection with a bounding box

Triangle similarity was utilized to calculate the distance between the detected vehicle and subject vehicle as mentioned in equation (1). After calculating the distance between two vehicles, the distance difference between frames could be calculated. Finally, the speed of the vehicle is calculated using the frame rate.

D. *Web API development*

Python Web API was selected as the front end for this research because it could be accessed by any application or device. Firstly, two Hypertext Markup Language (HTML) pages were implemented as the Home page and Demo page. The Home page includes a form that contained spaces for the focal length of the front view camera and web API Uniform Resource Locator (URL) which locates video stream from the camera.

The real-time video streaming by processing the input video stream from the front view camera was included in the Demo page. An alert was generated as an alarm sound when there is any danger present while overtaking another vehicle. Also, the distance from the subject vehicle to the detected vehicle and speed of the detected vehicle are shown with the real-time video stream.

The session concept was used when passing input data from the Home page to the Demo page. Also, the render template option was utilized to pass input values when submitting the button pressed by the user.

The frames that belonged to the input video transferred through road marking detection, traffic sign detection and finally, vehicle detection pipelines. Then Output frames were resized to fit into the HTML page and frames were encoded to bytes. Then those bytes were included in a NumPy array and later transformed into a byte array. Finally, the stream with context method streaming was performed.

In the flask app.py, template rendering was used to connect two HTML pages. The session was used to transfer data between two HTML pages. Stream with context method was used with mime-type as 'multipart/x-mixed-replace to stream the processed video output. The image of the web API is shown in Fig.8 below.
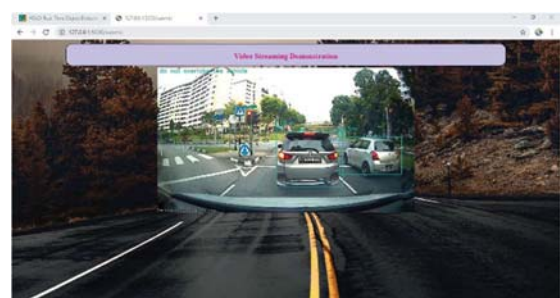

Fig. 8. Python web API video demonstration page

IV. RESULTS

The automated Vehicle Overtaking System has four main sections. Even though the final section (Python web API) displays only the output of the system, the other three sections provide predictions and detections for the outcome.

In the Road markings detecting section, the main goal is to detect the solid lines on the road and informing the driver 'not to overtake at this moment'. The system does not detect broken lines on the road. If there is a broken line, the system does not generate any alert sign because it is safe to overtake the vehicle in terms of lane line rules. The proposed system with broken lines is shown in Fig. 9.

28

Fig. 9. Road with broken lines

If there is a solid line, then the system detects it and gives an alert to the driver that does not overtake now. The proposed system with a solid line is shown in Fig.10.

Fig. 10. Solid line detection

The accuracy levels of road markings detection in straight roads and bent roads are shown in the following table.

TABLE I.   ROAD MARKINGS DETECTION ACCURACY

Road Feature	Total Frames	Detected Frames	Accuracy Percentage
Bend Road	102	88	86.27
Straight Road	110	102	92.73

The second section of the system addresses the traffic sign detection. The system detects pre-trained traffic signs such as pedestrian crossing, no entry and etc. In this study, the system was trained on traffic signs which are relevant for overtaking. If the system detects a 'trained traffic sign', the system warns the driver. The proposed system of a traffic sign detected occasion is shown in Fig.11.

Testing was done for two traffic signs such as pedestrian crossing and stop. 15 frames were used for two cascade classifier files respectively. Cascade Trainer GUI was used to testing tasks [10]. The accuracy of traffic sign detection is given in Table II.

Fig. 11. Traffic sign detection

TABLE II.   ACCURACY OF HAAR CASCADE CLASSIFIERS

Cascade.xml file	Correctly detected frames	False detected frames	Detection rate (%)
Pedestrian crossing	12	3	80
Stop	14	1	93.33

The final step of the proposed system is vehicle detection and predicting overtaking possibilities according to certain conditions. All the vehicles on the frame are detected and then the distances from the camera to those vehicles were calculated and displayed on the frame. Calculated distances were used to generate alerts on vehicle overtaking. A clear road occasion is shown below.

Fig. 14. Clear road and safe

If the distance between two vehicles is less than 15 meters then the system starts to check availability for overtaking. Also, if there is an oncoming vehicle then distance should be 30 meters or more between oncoming vehicle and front vehicle. If all the conditions are satisfied, the system considers it is safe to overtake and give an alert to drivers as "safe" displayed on the video stream. Otherwise, the proposed system alerts the driver as "do not overtake" displayed and the reason is also displayed. The following Fig.15 shows an overtake occasion that alerts the driver as not to overtake with the reason.

29

Fig. 15. Vehicle detection & predicting

This proposed system was used previously trained weight files and configuration files of the Yolov3 network for vehicle detection.Yolov3 has the most accuracy and speed among other networks. Yolov3 has 57.9% accuracy in mAP and also its accuracy is measured as 0.5 Intersection Over Union (IOU). The performance of yolov3 is graphed in the below image [11].
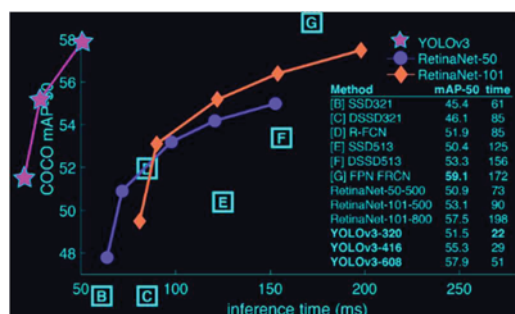

Fig. 16. Performance of yolov3 compared with other detectors.

V. CONCLUSION

The proposed system can generate alerts to warn drivers when overtaking based on road markings, traffic signs and distances between vehicles.

The method of triangle similarity was used in distance calculating. The perceived width of the vehicle was used as the width of the bounding box. There can be an error in distance calculation if the bounding box is not fitted to the detected vehicle. It can cause an error in the speeds of vehicles as well. The system was better able to detect road markings in straight roads rather than bend roads.

On some occasions, there can be inaccurate detections when the system performs traffic sign detection on input frames. As an example, when the system detects school and pedestrian crossing traffic signs, there can be wrong detections. The reason for that can be both two traffic signs include persons. More intensive training of the system on traffic signs may address this issue.

Also, while the subject vehicle is overtaking another vehicle, the front view camera can be focused on the other lane. Until the vehicle comes back to the previous lane, some detections are not valid. As an output of this model, it gives an alarm sound when there is a danger. A beep sound was embedded in the video stream that comes after processing input frames. There can be a very small time gap (less than second) between actual danger time and beep sound time. This error can be fixed by introducing a hardware component for this task.

If a vehicle is integrated with an early warning system such as one proposed it can detect dangerous and illegal actions while driving, and generate alerts. Python, OpenCV, and Flask are the main technologies used for the development of this system. Any device or application can access this system using the Python API.

To utilize commercially, the accuracy of the distance calculation, speed detection, traffic sign detection and road markings detection needs to be improved further. Also, the system needs to be trained for all the traffic signs. The efficiency of the system can be increased by optimizing algorithms.

Ideally, all the output data must be converted into sound notifications as the drivers are not able to read the information displayed while driving.

The system assumed widths of all vehicles as a constant value. The research work can be further improved by adding capabilities to detect the width of the vehicle as well. This will result in more accurate predictions.

REFERENCES

[1] R. Hadi, G. Sulong and L. George, "Vehicle Detection and Tracking Techniques: A Concise Review", Signal & Image Processing : An International Journal, vol. 5, no. 1, pp. 1-12, 2014. Available: 10.5121/sipij.2014.5101.

[2] A. Ousuf, N. Ali, S. Anwar And T. Das, "Development of a Safe Overtaking System according to Bangladesh Highway Road", 4th International conference On Mechanical Engineering And Renewable Energy Icmere, 2017.

[3] Gholami, Arash & Dehghani, Abbas & Karim, Mohamed,"Vehicle speed detection in video image sequences using CVS method",. International Journal of Physical Sciences. 5,2010

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767, 2018.

[5] Bhaskar, Prem & Yong, Suet,." Image processing based vehicle detection and tracking method. "1-5. 10.1109/ICCOINS.2014.6868357,2014

[6] A. Rosebrock,2019 "Find distance from camera to object using Python and OpenCV." [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/ [Accessed 26 Nov. 2019].

[7] Low, Chan & Zamzuri, Hairi & Mazlan, Saiful,." Simple robust road lane detection algorithm." 1-4. 10.1109/ICIAS.2014.6869550.,2014

[8] P. Prabhu, "Digital image processing techniques – a survey", Golden Research Thoughts, 2016.

[9] Kanburoğlu, A. Buğra & Tek, F.A Haar ,"Classifier Based Call Number Detection and Counting Method for Library Books.",10.1109/UBMK.2018.8566314,2018

[10] Amin,"Cascade Trainer GUI - Amin", [Online]. Available: http://amin-ahmadi.com/cascade-trainer-gui/. [Accessed: 23- Dec- 2019]

[11] J. Redmon. (2019). YOLO: Real-Time Object Detection. [online] Pjreddie.com. Available: https://pjreddie.com/darknet/yolo/ [Accessed 18 Dec. 2019].