ISSN 2409-2665
Journal of Logistics, Informatics and Service Science
Vol. 9 (2022) No.3, pp.208-225
DOI:10.33168/LISS.2022.0315

Forward Collision Warning for Autonomous Driving

Jun Jie Wee, Connie Tee + and Michael Kah Ong Goh

Faculty of Information Science & Technology, Multimedia University, Melaka, Malaysia

+*tee.connie@mmu.edu.my (corresponding author)*

Abstract. Reckless driving poses great danger to users and vehicles on the road. Studies have shown that reckless driving accounts for 60% of traffic accidents every year. Reckless driving can be caused by various factors including wine racks, drag racing, sleep deprivation and inexperienced driving. Due to these reasons, autonomous driving has received immersive attention in in the recent years. Forward collision warning is one of the core safety components in the development of autonomous vehicle. A forward collision warning system issues an early warning when a potential collision is detected in front of the ego vehicle. This paper presents a pipeline approach for visual-based forward collision warning. Deep learning-based object detection and lane detection modules are integrated to sense the environment around the ego vehicle. If an object is sensed ahead of the vicinity of the ego vehicle, a warning will be triggered. A mean average precision 0.5 (mAP 0.5) of 37.2 has been achieved with the proposed method. Empirical tests show that the proposed approach can work well with different road conditions including straight and curved roads, junctions, as well as different times of the days (e.g. days and nights).

Keywords: autonomous driving, object detection, lane detection, computer vision

1. Introduction

In 2009, Google initiated the autonomous driving project. Autonomous driving technology has been constantly advancing thereafter. Autonomous driving has gradually entered into people's life after Tesla came out with their first version of AutoPilot model known as Model S in October 2014. In year 2021, autonomous driving had reached level 3 out of 6 levels in driving automation where level 0 to level 5 signifies fully manual to fully automated, respectively. It is estimated that full automation will be released by 2025. The current autonomous driving systems still encounter much false detection, especially on roads without lane markings and low-light conditions (Haixia et al., 2021; Chen et al., 2020). Therefore, there is a need for safety mechanism like collision warning for autonomous driving.

There are many advantages with autonomous driving. An autonomous driving vehicle can automatically detect objects on the road and avoid possible collision. Hence, it would greatly reduce the risk of accidents on the road. Every year, there are 1.17 million deaths around the world caused by road accidents, of which 3 out of 5 people are pedestrians, 2 out of every 5 people are children. More than 70% of the causes of road accidents are due to drivers' negligence where the most serious cause is reckless driving.

A forward collision warning system is one of the critical components in autonomous vehicle safety measure. In the market, forward collision has been around for some time and most of the forward collisions warning systems are using sensors to make detection. There have been many manufacturers that produce sensors to detect objects on the road and notify users to avoid crashing such as Bosch, Continental AG, ZF-TRW and Autoliv.

In this study, we present a forward collision warning system for autonomous vehicles by using visual approach. The research is divided into two parts which are lane detection and object detection. Lane detection is used to estimate whether the ego vehicle is in the correct lane on the road. On the other hand, object detection helps to calculate and detect safety zones in different forward angles and distances between vehicles. Both of the components are implemented via a deep pipeline approach. Experimental results show that a mAP 0.5 of 37.2 can be achieved using the proposed approach. Empirical tests have also testified the robustness of the proposed system against different illumination factors and road conditions.

2. Related Works

2.1. Object detection

2.1.1. Conventional approaches for object detection

In 2018, Raghunandan et. al (2018) proposed an object detection algorithms for video surveillance applications. The authors worked on object detection using colour, skin and face information. In the study, several techniques such as the Viola Jones algorithm was applied to detect the facial feature. On the other hand, the YCbCr model (Green (Y), Blue (Cb), Red (Cr)) was used to detect the skin, while LIBS (Laser-Induced Breakdown Spectroscopy) was used to detect colour. The study faced challenges in LIBS (Laser-Induced Breakdown Spectroscopy) because the model failed to detect small objects such as leaves in the background. The study reported an accuracy of 95% using the proposed technique.

Sengar et al. (2016) presented a moving object detection method based on block-based frame differencing. As a moving object has high optical flow and noise corruption which affected real-time object detection, the study resorted to detect the object in videos by moving object detection approaches with inter-frame differencing and three-frame differencing. The proposed method allowed two video frames to identify the background and foreground pixels and solved the problems of aperture and ghosting. The proposed approach had good integrity of the objects in a different complex environment with a noisy background. A low error rate of 10.36% was reported in the study.

In addition, Sumati et. al (2016) proposed a human object detection method by using Histogram of Gradient (HOG), Histogram of Bar (HOB), Histogram of Colour (HOC) and Block Orientation (BO) features. In HOG, each cell feature of four directions were normalized after being summed to reduce the data dimension. HOC was applied to allow distinguishing mixture of pure colour information and intensity information. On the other hand, HOB helped to model an object into bar and blobs while BO helped to reduce false object detections. In the experiments, a F-measure with 41% was achieved by the HOG + BO features and HOG, HOB, HOC features.

2.1.2. Deep learning approaches for object detection

Kim et. al (2018) introduced an Object Bounding Box-Critic Networks for occlusion-robust object detection in road scene. The study worked on occlusion in road scenes. Occlusion affected the stability of object detection. Different obstacles were tested in the study, such as item occlusion, huge scale variations, and so on. Techniques such as actor-critic network were utilized to achieve robust object detection in occlusion. A multiple critic network was also presented to perform Bounding Box prediction (BB map), and a mix generative adversarial networks (GAN) with numerous actor-critic networks was presented to improve performance. On the feature encoding part, a VGG16 object detection network was developed. After that, Faster RCNN was used to perform object detection on the KITTI dataset. The proposed method had the best performance on object detection in level easy, medium and hard in three types of objects which is pedestrian, car and cyclist. Precision-recall curve (AUC) of (0.862, 0.695, 0.655), (0.950, 0.910, 0.862) and (0.730, 0.666, 0.630) was reported. The study concluded that the OBB-Critic network was one of the best solutions to detect object occlusions.

Kim et al. (2019) proposed an Attentive Layer Separation for object classification and object localization. The attention network and ResNet-101 were used as the backbone of the network. Both attention maps were generated to input the layer separation part and performed the two tasks. A mean average precision with 77.2 mAP was reported for Faster R-CNN and 80.1 mAP for attention network. The study found that attention network was useful to detect object occlusions.

On the other hand, Sai et al. (2019) proposed object detection and of object counting approach using a Tensor Flow Object Detection API. The models were trained using Faster RCNN (faster Region-based Convolutional Neural Network) to help define the number of objects in the image. In the experiment, an accuracy of 81.81% was achieved by the model. The study found that better performance could be attained by having more training data.

Alternatively, Wang et al. (2019) presented an object detection method with deep learning for underwater environment. The study worked on detecting living things such as fish in an underwater environment. You Only Look Once (YOLOv3) which had good bounding box predictions and good class predictions was utilized in the study. Feature Pyramid Network supported and Darknet-53 supported frameworks were adopted in the experiment. The study obtained almost 0.95 Intersection over Union (IOU) and almost 0.05 losses. It was observed that the accuracy would increase to a certain extent and it would be balanced after that.

2.2. Lane detection

2.2.1. Conventional approaches for lane detection

Deng et al. (2018) proposed a double lane line edge detection method based on Constraint Conditions Hough Transform. They investigated lane identification using the constraint Hough transform double-edge extraction approach. The approach was used to convert the lane line area into red green blue (RGB) and gained the image's edge using Canny operator. The restriction of the straight lane line was detected using Hough transform. The curve lane was finished using least-squares fitting. The study got a 98.5% accuracy/recognition rate after the experiment and the authors pointed out that the method would shorten the processing time by using polar angles of Hough transform.

Recently, Swetha et al. (2021) proposed a Shape Supervised Learning Algorithms (SSLA) based traffic sign and lane detection method for autonomous cars. The research was performed with SVM to train the model. SVM was used to identify the appropriate shape and adopted the Hough line transformation techniques. By using the SSLA, accuracy of SSLA was almost 0.95. The study succeeds in detecting the lane line and showed that SSLA was useful for enhancing the safety of autonomous cars.

Apart from that, Bhupathi et al. (2020) proposed an augmented sliding window technique to improve detection of curved lanes for autonomous vehicles. In the research, the sliding window approach and lane-fitting algorithm were used. Image operations such as colour-based extraction, grayscale image conversion, edge detection and perspective transformation were also applied. The Sobel edge detector was used to find the starting position. After that, gradient direction and grey level intensity and canny edge detection were used to segment dominant pixels. An accuracy of 96.26% was reported for the lane detection approach. The proposed method showed good performance to deal with sharp curve and dashed lines.

In addition, Zhu et al. (2021) proposed a moment-based multi-lane detection and tracking algorithms. They applied several methods such as Kalman filtering and state-of-the-art neural networks, and combined them in the research. The proposed method aimed to detect multi-lane and curve lanes. Multi-lane detection was separated into two steps which determined the starting point and derived dynamic Return on Investment (ROI) to extract lane segments. An accuracy rate of 98% was obtained by using the proposed method.

2.2.2. Deep learning approaches for lane detection

Li et al (2021) proposed a flexible lane detection method using Convolutional neural network (CNNs). Self-encoders and decoder networks (AE) and convolution neural networks were studied in the research. The encoder network was used to perform image feature extraction and representation. The decoder network performed pixel-level fine-tuning by combining deep and shallow semantic information. An accuracy of 96% was achieved. The study found that AE had a good performance on lane boundary recognition and classification tasks for complex lane scenes and the proposed model met the requirements of real-time applications.

Chen et al. (2020) proposed a non-local spatial information based lane detection method. The authors researched on non-local partial information modules which were composed of Secure Convolutional Neural Network (SCNN) modules and asymmetric non-local modules. VGG16 was used as the input and the top hidden layer. The non-local partial information module was divided into upward, downward, leftward, and rightward. In the study, Fl-measure of 73.9% was obtained. The module was validated by CULane and achieved an excellent result in the driving environment.

Zou et al. (2020) proposed a robust lane detection from continuous driving scenes using deep neural networks. They applied Deep convolutional neural network (DCNN) and Deep Recurrent Neural Network (DRNN) in the research. The encoder-decoder architecture was used, and ConvLSTM was applied to handle the encoder feature of the inputs. The result returned a high accuracy of 98%. The study claimed that this was one of the best methods for lane detection.

Besides, Neven et al. (2018) proposed an end-to-end lane detection method using instance segmentation. In this study, they used the LaneNet and H-Net. LaneNet could identify and cope with lane changes and could cluster loss functions for one-shot segmentation challenges. H-Net was also utilized to tailor the loss function and optimized it from beginning to finish in order forecasting the parameters of a viewpoint transformation H. After testing, the authors discovered that their proposed approach had 96.4 percent accuracy and could detect lanes in 50 frames per second.

Qian et. al (2020) proposed a deep learning transmittance network (DLT-Net) for joint detection of drivable areas, lane lines, and traffic objects. They used the DLT-Net unified neural network to recognise drivable zones, lane lines, and traffic objects in their study. They also used encoders and decoders to extract rich visual information in the traditional way. A score of 68.4 percent accuracy was reported in the study. Nonetheless, various flaws were discovered in the study, like non-lane region being misclassified as part of the drivable region, and the inability to adjust to extreme reflecting situations, and also not being unable to forecast broken lane lines properly.

3. Proposed Solution

3.1. Deep learning-based object detection and lane detection

In this study, a deep learning approach is used to perform object detection and lane detection. Specifically, the YOLACT algorithm is applied. YOLACT is an improvement over its predecessor, YOLO (Redmon et al., 2016). YOLACT has the same high performance as YOLO, but YOLACT has better flexibility and performance speed than YOLO. YOLACT has been developed to focus on improving the speed performance without losing accuracy. The main aim is to increase its flexibility to enable real-time predictions.

YOLACT can predict high-quality segmentation masks of objects and describe their shapes like MASK RCNN, which further improves the flexibility that YOLO does not have. To generate real-time predictions, YOLACT uses ResNet-101 or ResNet-50 to create convolutional image pyramids to increase computational speed. After generating multiple regular convolutional image pyramids, Protonet and NMS help YOLACT generate prototype and mask coefficients, respectively. Finally, it would combine Protonet and NMS to crop and threshold the image to generate object and mask detections. The quality of the mask generated by YOLACT have higher quality than other algorithms and the mask is very close to the object.

In this paper, two models for YOLACT are developed, one for object detection and the other for lane detection. A pre-trained model built using the Common Objects in Context (COCO) dataset is used for object detection. On the other hand, the model for lane detection is trained using a self-collected dataset. Images containing different types of road lanes, e.g. straight lane, curved lane, junctions, lanes with different lighting conditions (e.g. bright lighting, medium lighting, dark lighting, shadows) are captured. The lane markers are manually annotated on the images. These annotated images are then used to train a custom YOLACT model for lane detection.

3.2. Forward collision warning

The proposed forward collision warning module receives video input from a camera. The camera can be any smart devices like IoT sensors that are mounted at the front of the ego vehicle. The overall process flow of the proposed system is presented in Figure 1.

At the start of the program, the proposed system checks for video capture and if no video is detected, it ends the process. If a video is captured, it would generate frames from the video to detect objects from each frame via YOLACT. The system removes objects with scores over 0.6 from the frame and loops through to generate recovery object frames. At every 20 iterations, lane detection is performed by using YOLACT and the system verifies if any lane is detected in the frame. If no lane is detected, it continues with the next frame until a lane is detected. If a lane is detected in the current frame but no frame is detected in the next frame, the lane landmarks detected in the previous frame will be used. Otherwise, if more than lanes are detected, the highest lane detection score will be used to generate lane masks to check for overlapping objects and lane masks. If an overlapping objects and lane masks are detected in the frame, an alert will be triggered. Figures 2 and 3 depicts the output of normal and output with warning message for the proposed forward collision warning system.

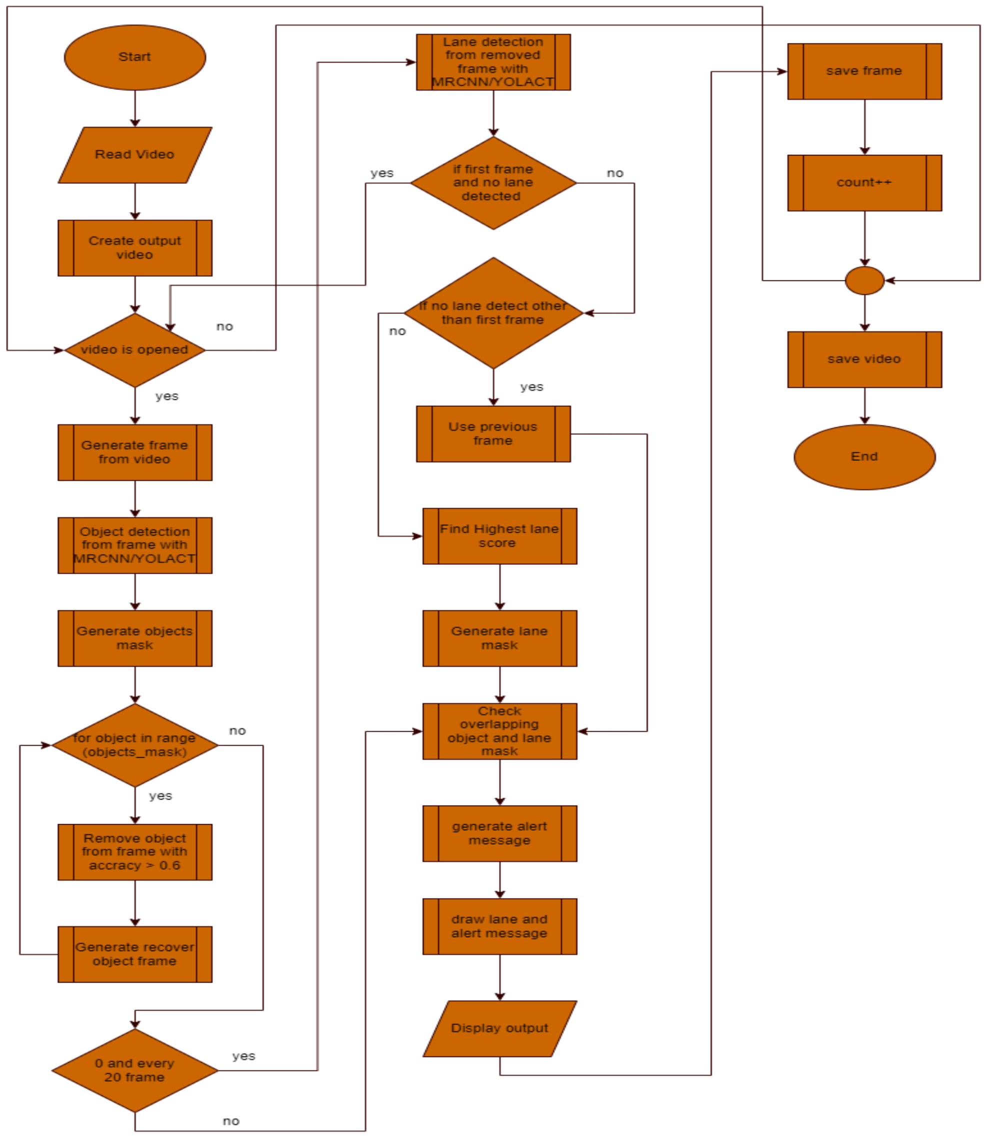

Fig. 1: Block process flow.

Fig. 2: Output for a safe driving region.

Fig. 3: Issue warning for objects detected in safe driving region.

4. Experimental Results

4.1. Object detection

First, a number of deep learning-based object detection methods have been explored. They include YOLO, YOLACT and Mask RCNN. The COCO pre-trained weights are used in all the models.

The reason why the processing speed of Mask RCNN is prolonged is that the detection mask itself takes a long time, not to mention that it often has a lot of false detections on the mask, resulting in many objects being redundant on the mask, so its speed will be serious slower. Compared with MASK RCNN, YOLACT's fps will be faster, because YOLACT's object detection is more accurate than MASK RCNN so the false detection object will be reduced, resulting in YOLACT having the fastest detection speed. YOLOv5 is the fastest because it only detects the bounding box of the object and does not spend time sketching the object's mask.

The final conclusion would explain their pros and cons when applying this system. Table 1 summarizes the performances of using the different object detection approaches. Mean average precision is a method to calculate a conditional probability of witnessing data given a model weighted by a previous probability or belief about the model; while intersection over a union (IOU) indicates how much the expected and ground truth bounding boxes overlap. Frames per second (fps) indicates how many frames can be generated in each second from the system. The speed for the different methods are also provided in Figure 4. Among the methods, YOLACT and Resnet-101 offer favourable results because they flexible, accurate and real-time performance.

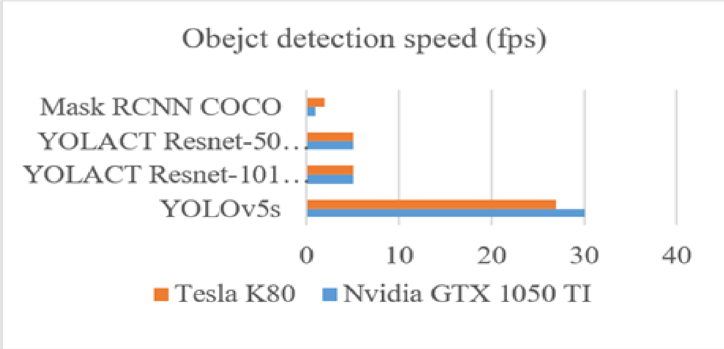

Fig. 4: Object detection speeds.

In Table 1, the mean average precision is the method to calculate a conditional probability of witnessing data given a model weighted by a previous probability or belief about the model and the intersection over a union (IOU) indicates how much the expected and ground truth bounding boxes overlap. Frames per second (fps) is a method to calculate how many frames can be generated in each second from the system.

Table 1. A comparison of different object detection approaches

<table>
<tr><th rowspan="2">Methods</th><th rowspan="2">Mean average precision (mAP) IoU = 0.5</th><th rowspan="2">Advantage (Dwivedi 2020; YOLACT; Mask R-CNN 2021)</th><th colspan="2">Frame per second (fps)</th><th colspan="2">Performance</th></tr>
<tr><th>Nvidia GTX 1050 TI</th><th>Tesla K80</th><th>Pros</th><th>Cons</th></tr>
<tr><td>YOLO YOLOv5s</td><td>37.2</td><td>Better performance in detecting smaller objects, no overlapping boxes</td><td>30</td><td>27</td><td>Fastest processing speed, detection is accurate</td><td>Not flexible</td></tr>
<tr><td>YOLACT Resnet-101 COCO</td><td>29.5</td><td rowspan="2">Higher Quality of Masks, High speed, flexibility</td><td>5</td><td>5</td><td rowspan="2">Flexible, detection is accurate, faster processing speed</td><td rowspan="2">Slower prediction speed than YOLO</td></tr>
<tr><td>YOLACT Resnet-50 COCO</td><td>27.0</td><td>5</td><td>5</td></tr>
<tr><td>Mask RCNN COCO</td><td>29.6</td><td>Simple to train, good performance, flexibility</td><td>1</td><td>2</td><td>Flexible</td><td>Slowest prediction speed, annotation shape is not smooth, detection is not accurate</td></tr>
</table>

4.2. Lane detection

4.2.1. Dataset preparation

Instead of using a pre-trained model, the lane detection model is trained using a custom dataset. For this purpose, images of different lane conditions are captured. A Huawei Nova 5t hand phone is used as the recording device. The device is placed in front of the vehicle to avoid unnecessary jitters in the video quality. The recording starts when the car starts to move. In total 25 videos have been collected, thirteen morning videos, four evening videos, and eight raining videos that vary in length, driving conditions, and environments are acquired to ensure their uniqueness. In this way, we can ensure that the trained model is able to detect lanes in different driving conditions and situations. Some sample lanes conditions are portrayed in Figure 5. After that the videos are cropped to 1980x1080 pixels. Makesense AI is used to turn each frame into its own bounding boxes and polygon annotations and the annotations and bounding boxes are exported into a COCO.json file. Figure 6 illustrate the annotated frame with bounding box.

Fig. 5: Frames for different lane conditions. top left: night; top right: junction; bottom left: curved lane; bottom right: straight lane.

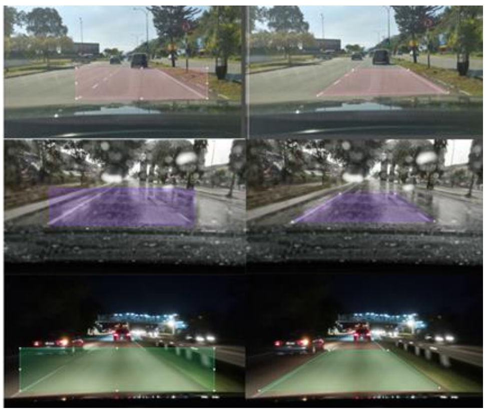

Fig. 6: Frames annotated with bounding box and polygon.

4.2.2. Lane detection evaluation

A number of models have also been investigated for lane detection. The models tested include conventional YOLO, YOLACT and Mask RCNN. The performances of the different lane detection approaches are shown in Table 2.

The reason why the processing speed of Mask RCNN is prolonged is that the detection mask itself takes a long time, not to mention that there is much false detection on the mask, resulting in many redundant objects detected on the mask. This makes the speed much slower. As compared to MASK RCNN, YOLACT's fps is faster. YOLACT's object detection rate is more accurate than MASK RCNN and false detection is reduced. YOLOv5 is the fastest because it only detects the bounding box of the object and does not spend time sketching the object's mask. Nevertheless, the quality of the mask generated is less superior.

The speed comparisons among the methods are also illustrated in Figure 7. We observe that YOLACT gives the best performance in terms of speed and accuracy in the test. Therefore, the YOLACT model is adopted in the subsequent evaluations.

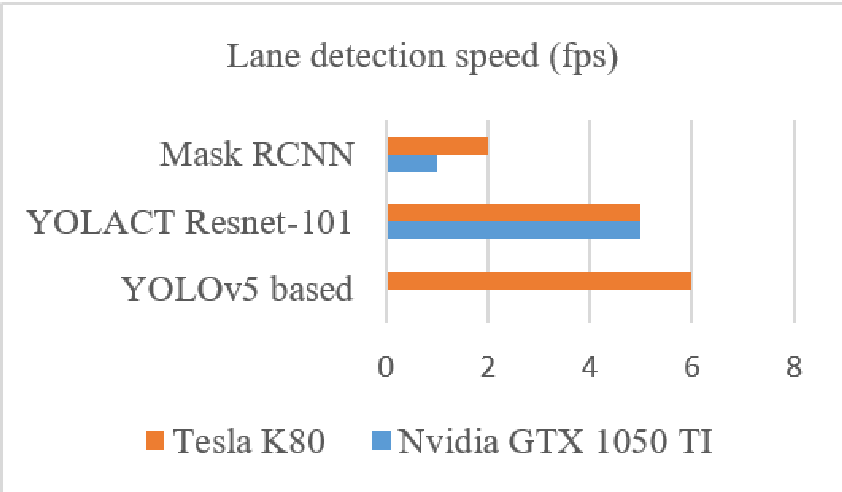

Fig. 7: Frames annotated with bounding box and polygon.

In Table 2, the mean average precision is the method to calculate a conditional probability of witnessing data given a model weighted by a previous probability or belief about the model and the intersection over a union (IOU) indicates how much the expected and ground truth bounding boxes overlap. Frames per second (fps) are a method to calculate how many frames can be generated in each second from the system.

Table. 2: Comparison of lane detection approaches.

Method	Mean average precision (mAP) IoU = 0.5	Advantage (Dwivedi 2020; YOLACT; Mask R-CNN 2021)	Frame per second (fps)		Performance	
			Nvidia GTX 1050 TI	Tesla K80	Pros	Cons
YOLOv5 with conventional method	-	Better performance in detecting smaller objects, no overlapping boxes	-	6	Fastest processing speed	Not flexible, detection is inaccurate
YOLACT Resnet-101	24.56	Higher Quality of Masks, High speed, flexibility	5	5	Flexible, detection is accurate, faster processing speed	Slower prediction speed than YOLO
Mask RCNN	98.2	Simple to train, good performance, flexibility	1	2	Flexible	Slowest prediction speed, annotation shape is not smooth, detection is not accurate

4.3. Hardware evaluation

A forward collision warning system must operate under different conditions. Therefore the system needs to operate very at top speed. GPUs play a critical role in supporting the speed of the system. So it is important to evaluate the performance of the GPU. In this study, three types of GPUs have been tested namely NVIDIA GTX 1050 TI, NVIDIA Tesla K80 and NVIDIA RTX 2070s, as this are the GPUs we currently have. Both NVIDIA GTX 1050 TI and NVIDIA Tesla K80 have an object detection rate of 5fps. This is important to support the execution of the system, to not only detect but also generate masks for lanes and objects, check for objects ahead and generate alert messages.

A forward collision warning system must operate under different conditions. Therefore the system needs to operate very at top speed. GPUs play a critical role in supporting the speed of the system. So it is important to evaluate the performance of the GPU. In this study, three types of GPUs have been tested namely NVIDIA GTX 1050 TI, NVIDIA Tesla K80 and NVIDIA RTX 2070s, as this are the GPUs we currently have. Both NVIDIA GTX 1050 TI and NVIDIA Tesla K80 have an object detection rate of 5fps. This is important to support the execution of the system, to not

only detect but also generate masks for lanes and objects, check for objects ahead and generate alert messages.

When running real-time test, NVIDIA GTX 1050 TI or NVIDIA Tesla K80 could not perform well as the processing time is around video time *~10. Only NVIDIA RTX 2070s can achieve timely detection. This processor is 2.87 to 4.64 times faster than NVIDIA GTX 1050 TI. A comparison of the different hardware is provided in Table 3.

Table. 3: Comparison of using different hardware.

GPU	NVIDIA RTX 2070	NVIDIA GTX 1050 TI	Tesla K80
Avg. Locally-deformable PRT (Bat)	129 fps	38 fps	
Avg. High dynamic range lighting (Teapot)	120 fps	41.7 fps	
Avg. Render target array GShader (Sphere)	175 fps	37.7 fps	
Avg. NBody particle system (Galaxy)	126 fps	39.5 fps	
Processing time	(video time * ~3)	(video time * ~10)	
Advantage (GeForce GTX 1050; GeForce RTX 2070 vs Tesla K80)	High value for money, highest clock speed	Cheap	Highest value for money, Highest memory, cheapest

4.4. Real-time tests

The proposed forward-collision warning system is tested with real-time input videos. Different lane types and road conditions have been tested. The output for the different tests is illustrated in Figures 8 to 15. In general, we observe that the proposed method can work very well in dealing with the different lane types like straight lane, curve lane and junction. It can also detect the objects and lane correctly under different weather condition and different times of the day, e.g. morning, night and rainy day.

While the forward-collision warning system works well, it suffers from some detection errors sometimes. Figure 16 shows a false detection output at the high deceleration zone. When the car passes through the high deceleration area, the camera will move upward, and whole screen is mistakenly detected as the lane. Figure 17 shows false detection of a vehicle. This error happens because the detected object's score is below 0.6, so the fore-front vehicle is not detected as an obstacle in front of the ego vehicle.

Fig. 8: Straight lane in the morning.

Fig. 9: Straight lane in the night.

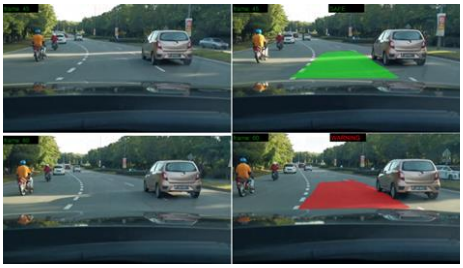

Fig. 10: Curved lane in the morning.

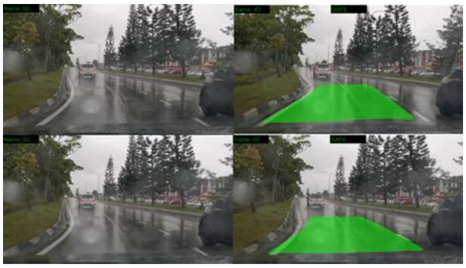

Fig. 11: Curved lane during rainy day.

Fig. 12: Raining day in the morning.

Fig. 13: Junction in the morning.

Fig. 14: Junction in the night.

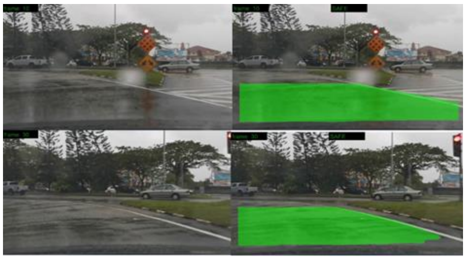

Fig. 15: Junction during rainy day.

Fig. 16: Output of misdetection lane at deceleration zone.

Fig. 17: Output of misdetection object.

5. Conclusion

Many lives have been lost due to reckless driving. Autonomous driving appears to be a promising solution to safe driving. In this paper, a visual-based forward-collision warning system is proposed for autonomous vehicles. The forward-collision warning system allows lanes and objects to be detected reliably from the road. In this research, different deep learning models have been tested for lane and object detection. Empirical tests show that the YOLACT technique yields the best performance. It has a processing speed of 5 fps and the object and lane masks predictions are very accurate.

In the future, efforts will be dedicated to explore more sophisticated lane and object detection techniques. The fusion of visual and sensor input will also be a potential research direction.

Acknowledgements

This project is supported by the IR Fund (Grant no. MMUI/220041). The authors would also like to thank the providers of the COCO dataset for sharing their databases.

References

Bhupathi, K. C. & Ferdowsi, H. (2020). An augmented sliding window technique to improve detection of curved lanes in autonomous vehicles. In *2020 IEEE International Conference on Electro Information Technology (EIT)*, 522–527. DOI:10.1109/EIT48999.2020.9208278.

Chen, H.-B., Pan, L., Lin, L.-J., & Huang, L.-Q. (2020). Non-local spatial information based lane detection. *In 2020 Cross Strait Radio Science Wireless Technology Conference (CSRSWTC)*, 1–3.DOI:10.1109/CSRSWTC50769.2020.9372627.

COCO - Common Objects in Context. https://cocodataset.org/#home (accessed Apr. 15, 2022).

Deng, G. & Wu, Y. (2018). Double lane line edge detection method based on constraint conditions Hough transform. In *2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, 107–110. DOI:10.1109/DCABES.2018.00037.

Dwivedi, P. (2020). YOLOv5 compared to faster RCNN. Who wins? Medium, Jun. 30, 2020. https://towardsdatascience.com/yolov5-compared-to-faster-rcnn-who-wins-a771cd6c9fb4 (accessed Jun. 04, 2022).

GeForce GTX 1050 Ti vs RTX 2070 [in 9 benchmarks]. https://technical.city/en/video/GeForce-GTX-1050-Ti-vs-GeForce-RTX-2070 (accessed Jun. 04, 2022).

GeForce RTX 2070 vs Tesla K80 [in 2 benchmarks]. https://technical.city/en/video/GeForce-RTX-2070-vs-Tesla-K80 (accessed Jun. 04, 2022).

GitHub - haotian-liu/yolact_edge: The first competitive instance segmentation approach that runs on small edge devices at real-time speeds. https://github.com/haotian-liu/yolact_edge (accessed Apr. 06, 2022).

Haixia, L. & Xizhou, L. (2021). Flexible lane detection using CNNs. In *2021 International Conference on Computer Technology and Media Convergence Design (CTMCD)*, 235–238. DOI:10.1109/CTMCD53128.2021.00057.

Kim, J. U., Kwon, J., Kim, H. G., Lee, H., & Ro, Y. M. (2018). Object bounding box-critic networks for occlusion-robust object detection in road scene. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 1313–1317. DOI:10.1109/ICIP.2018.8451034.

Kim, J. U. & Man Ro, Y. (2019). Attentive layer separation for object classification and object localization in object detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, 3995–3999. DOI:10.1109/ICIP.2019.8803439.

Mask R-CNN: A beginner's guide. viso.ai, Mar. 19, 2021. https://viso.ai/deep-learning/mask-r-cnn/ (accessed Jun. 04, 2022).

Mask R-CNN for Object Detection and Segmentation. Matterport, Inc, 2022. Accessed: Apr. 06, 2022. [Online]. Available: https://github.com/matterport/Mask_RCNN.

Neven, D., Brabandere, B., Georgoulis, S., Proesmans, M., & Van Gool, L. (2018). Towards end-to-end lane detection: An instance segmentation approach. 291. DOI:10.1109/IVS.2018.8500547.

Qian, Y., Dolan, J. M., & Yang, M. (2020). DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects. *IEEE Trans. Intell. Transp. Syst.*, 21(11), 4670–4679. DOI:10.1109/TITS.2019.2943777.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. DOI:10.1109/CVPR.2016.91.

Raghunandan, A., Raghav, M. P., & Aradhya, H. V. R. (2018). Object detection algorithms for video surveillance applications. In *2018 International Conference on Communication and Signal Processing (ICCSP)*, 0563–0568. DOI:10.1109/ICCSP.2018.8524461.

Sai, B. N. K. & Sasikala, T. (2019). Object detection and count of objects in image using tensor flow object detection API. In *2019 International Conference on Smart*

Systems and Inventive Technology (ICSSIT), 542–546. DOI:10.1109/ICSSIT46314.2019.8987942.

Sengar, S. S. & Mukhopadhyay, S. (2016). A novel method for moving object detection based on block based frame differencing. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 467–472. DOI:10.1109/RAIT.2016.7507946.

Singh, S. S. & Gupta, S. C. (2016). Human object detection by HoG, HoB, HoC and BO features. In *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 742–746. DOI:10.1109/PDGC.2016.7913220.

Swetha, S. & Sivakumar, P. (2021). SSLA based traffic sign and lane detection for autonomous cars. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 766–771. DOI:10.1109/ICAIS50930.2021.9396046.

ultralytics/yolov5. Ultralytics, 2022. Accessed: Apr. 06, 2022. [Online]. Available: https://github.com/ultralytics/yolov5.

Wang, C.-C., Samani, H., & Yang, C.-Y. (2019). Object detection with deep learning for underwater environment. In *2019 4th International Conference on Information Technology Research (ICITR)*, 1–6. DOI:10.1109/ICITR49409.2019.9407797.

Zhu, D., Song, R., Chen, H., Klette, R., & Xu, Y. (2021). Moment-based multi-lane detection and tracking. *Signal Process. Image Commun.*, 95, 116230. DOI:10.1016/j.image.2021.116230.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2020). Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Trans. Veh. Technol.*, 69(1), 41–54. DOI:10.1109/TVT.2019.2949603.