EFFICIENT VEHICLE DETECTION AND TRAFFIC CONTROL MANAGEMEMT USING YOLOV5 - A SIMULATION-BASED APPROACH

Othman O. Khalifa[1,2*], Hariz Naufal Bin Mohd Daud [1], Muhammed Zaharadeen Ahmed [1], Aisha Hassan Abdulla[1], Abdelrahim Nasser Esgiar[3]

[1]Department of Electrical and Computer Engineering, Faculty of Engineering, International Islamic Unversity Malaysia, Malaysia
[2]Libyan Center for Engineering Research and Information technology, Bani walid, Libya
[3]Department of Electrical and Electronic Engineering, Sirte University, Libya

ookhalifa@gmail.com

Article history

Received: 16 Dec 2024

Received in revised form: 28 Dec 2024

Accepted: 30 Dec 2024

Published online: 31 Dec 2024

*Corresponding author: ookhalifa@gmail.com

ABSTRACT

Soon, the technology for self-driving vehicles will occupy the center stage in automotive engineering. Implementing such technologies can stimulate driving accuracy and reduce the rate of accidents caused by human error. Humans are prone to limitations such as fatigue, lack of focus, or boredom, which are significant contributors to accidents on roads and highways. This paper presents the feasibility of implementing smart cities to achieve efficient vehicle distinction and traffic control management using the concept of deep learning. This process involves training devices to adapt and exhibit the capability to distinguish realistic scene situations using simulation for validation. The YOLOv5 simulation tool is utilized due to its efficiency in generating the best Mean Average Precision (MAP). The findings reveal that the YOLOv5 model achieves a MAP of 60.36% with 50% recall and operates at 58 frames per second, demonstrating superior precision and efficiency compared to existing object detection models, making it a promising tool for real-life autonomous driving applications.

Keywords: Autonomous Driving, Deep *Learning, Smart Cities, Object Detection.*

1.0 INTRODUCTION

Self-driving technology is considered one of the most effective ways to reduce human driving errors. This idea is developed when researchers consider introducing advanced technology to assist humans in their driving roles. A digital device or a computer is not designed with limitations considering the concept of garbage in garbage out. Therefore, this technology implementation eliminates almost all the road mistakes by elimination cases of accidents and because of accumulated fatigue or negligence by humans [1].

For efficient smart cities, self-driving vehicles complement technology. This is due to the use of smart machines with more sensors and equipment's to achieve efficient performance. The two-way approach between the self-driving vehicles and data management technologically for smart cities would greatly enhance the present movements of vehicles. This involves the contribution of Researchers, manufacturers, and consumers to take great advantage of this technological evolution and ensure the world becomes a better place [2]. Also, to help in maintaining reliable electronic systems, researchers and technologists are doubling up efforts towards achieving tremendous progress in vehicular movements. This is practical in advanced countries and cities who are currently advancing in their technology. However, not only in limiting errors caused by humans' failure, but autonomous driving also helps decrease the rate of pollution because of improved driving and efficiency of fuel which makes regulation of traffic flow and parking problem less complicated [3].

2.0 LITREATURE REVIEW

2.1 The Autonomous vehicles

Technological advancement has pulled autonomous vehicles to the center stage of economic prosperity to many nations. Researchers are constantly working on possible ways of improving the autonomous automotive industry. Different technology entrepreneurs and companies are emerging because of the obvious future technology possess. This progress lies with a common goal of evolving the design plan of the autonomous vehicles. The advancement of autonomous vehicles has brought about enhancing enormous features such as lane maintenance guide, intelligent driving, autonomous braking, and parking. These features ensure an effective driving strategy. However, with this progress autonomous vehicle possess, the roads can not be declared efficiently safe for an individual driver to show zero conscience and concentration while driving. This is because computer machines and all smart electronic devices are required to engage in frequent maintenance procedures [4]. Experts in the field of autonomous driving has classified autonomous driving into six stages, starting from the stage 0 (which is strictly manual operation) to stage 5 (fully automated). Due to the limited and harsh operation rules of the autonomous vehicles of stage 3 features and above, the United States government has put a stop on its patronage. The tesla vehicles, owned by a multi billionaire named Elon musk, is one of the autonomous vehicles manufacturing companies which has constantly been producing self-controlled vehicles with standards and effective self-driving systems [5].

2.2 Region-based Convolutional Neural Network (R-CNN)

Detection of objects in accordance with deep learning concept to serve as a region based convolutional neutral network. This is referred to as regions with the CNN features (R-CNNs). The ideology behind R-CNN is centered around the proposals of the region. the four processes involved in the R-CNN [6]. These are: -

i. Region proposed ensures objects in the image.
ii. Extraction features obtained from the plan of region.
iii. The objects are arranged using the already extracted features.
iv. Vector machines are then used to modify the plans of the region.

At the initial stage, the inputs were arranged within the 2000 regions. This enables each of the regions to be put in the convolutional neutral network simultaneously. After mathematical computations the most accurate regions are then included into the neutral network. It generates bounding boxes independently to influence time limitations of the training model. Only a single region functions at a time with the neutral network. The timing of the YOLO models is relatively bigger and extensive. Figure 1 below, presents detection and classification conducted using the R-CNN model.

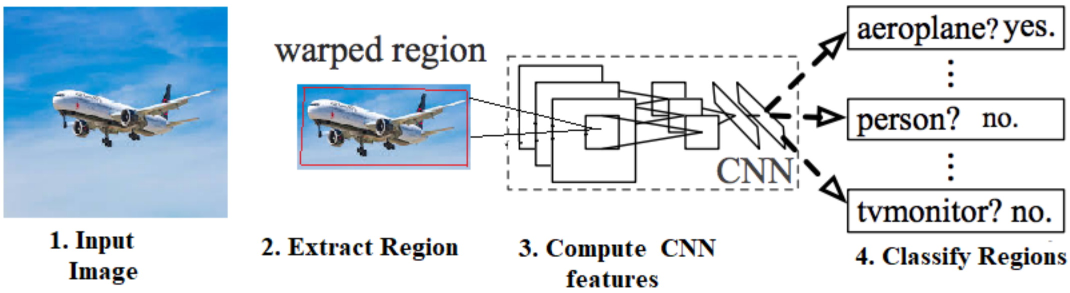


Figure 1: Flow of R-CNN network

2.3 Fast R-CNN

A major limitation of CNN is its enormous period taking during model training. However, this limitation is an advantage in the case of real time applications. This challenge stimulates implementing an algorithm for Fast R-CNN. The algorithm region's proposal has similar procedure with that of the CNN. The input image is aimed at obtaining a complex map feature for the purpose of identifying proposals of the regions. Regions that have already been identified are being warped in squares and then rearrange into a fixed size to ensure the layers are fully connected. The soft Max layer is used to forecast the category of a region proposal and the boxes bounding it off set values [7]. The fast R-CNN can be depicted in Figure 2 below.

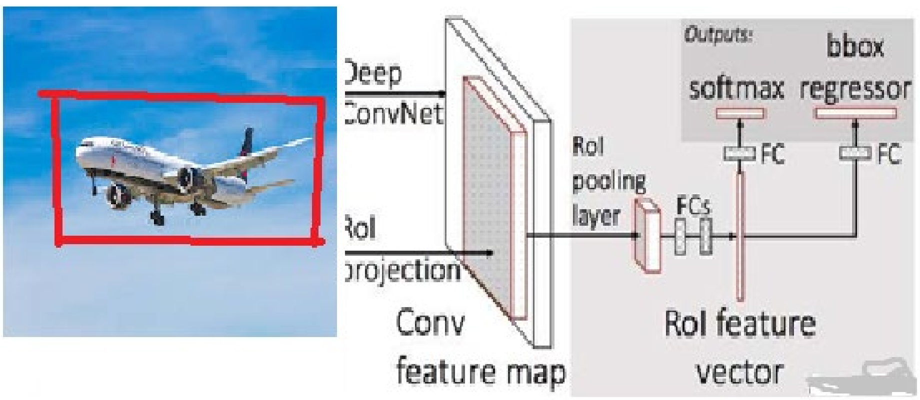


Figure 2: Flow of Fast R-CNN network

Due to the irrelevant feed of 2000 region proposal of the R-CNN, the Fast R-CNN is said to be more rapid. This is because the feature map of the Fast R-CNN has the capability of generating images upon a single glance at an object. Notwithstanding the fact that both algorithms operate on decisive search to obtain region proposals, the process is regarded as very slow and time consuming which is really a setback for an algorithm termed as the best, which makes it incompatible for real-time detection of objects during autonomous driving [8-9].

3.0 METHODOLOGY

This paper presents an enhanced approach of implementing network platform with deep learning algorithm using an effective sound system. This prototype is implemented with high efficiency and swift response for a real-time deployment in autonomous vehicles. The proposed flowchart presents the design, as illustrated in Figure 3.

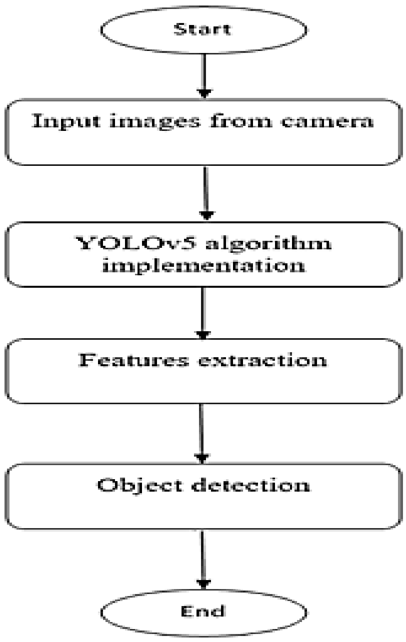


Figure 3: Flow of Proposed Solution

3.1 Training Object Detection Model with Yolov5

In the process of implementing the Udacity Self Driving Dataset, inserted videos generated with the aid of dataset are converted into frames. The dataset is then processed alongside the image for proper and detailed explanation. The dataset interpretation involves training exercises and feature extraction for enhanced accuracy. Establishing featured pyramid using YOLOv5 algorithm tool enables successful execution of all the epochs that the trained weight will then be transmitted into the directory.

PERINTIS eJournal

PERINTIS eJournal, **2024**, Vol. 14, No. 2, pp. 27-38

3.2 Dataset

The process of object classes arrangements to generate data pieces is termed dataset. The data set generated during our model training is being refined by cleaning the data pieces for easy recognition. The recognition process involves labelling the dataset immediately after it is subjected to practice on the field. Although the Dataset is being generated after the process of mathematical computations of images. This means image gathering procedure is required to achieve effective recognition.

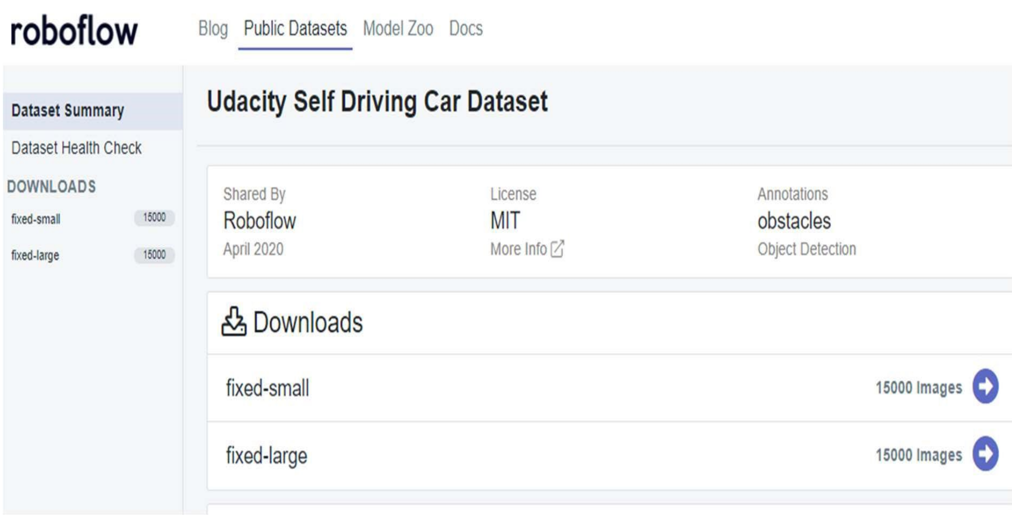


Figure 4: Udacity Self Driving Car Dataset by Roboflow

3.3 Dataset Preprocessing, Augmentation, and Challenges

The dataset was first cleaned to remove irrelevant images, corrupted files, and duplicates. Bounding boxes were carefully labeled to ensure accurate object detection. This involved manually correcting mislabeled or missing annotations from the original dataset.

Augmentation: Data augmentation techniques were applied to improve generalization, including random rotations, flipping, scaling, and brightness adjustments. These techniques helped simulate diverse real-world conditions.

Challenges: During training, resource constraints on GPU memory caused limitations in processing high-resolution images. Additionally, imbalanced data in certain object categories affected model performance, necessitating careful augmentation to address class imbalances.

However, Customizing or optimizing the YOLOv5 architecture for autonomous driving tasks typically involves tailoring the network to improve detection accuracy, speed, and reliability in dynamic and complex driving environments. Udacity dataset Augmenting with Driving-Specific Features with synthetic or real-world data that includes variations in weather, lighting, and occlusions (e.g., foggy or rainy conditions, night-time driving, or traffic-heavy scenes).

4.0 RESULTS ANALYSIS

In this paper, a comprehensive analysis of autonomous vehicles is conducted using the Udacity self-driving dataset to evaluate model efficiency. The study employs the YOLOv5 tool for simulation and image processing.

4.1. Performance Evaluation Metrics

To compute a model's performance rate, an analysis of accuracy based on True-Positive (TP), False-Positive (FP), and False-Negative (FN) values is required. Precision (PR) measures the model's ability to identify relevant objects and represents the percentage of correct positive predictions. Recall (RC) evaluates the model's ability to identify all relevant cases (ground truth bounding boxes), expressed as the percentage of correct positive predictions relative to all ground truths. The formulas for recall and precision are provided below: -

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + Fals\ Positrive} \tag{1}$$

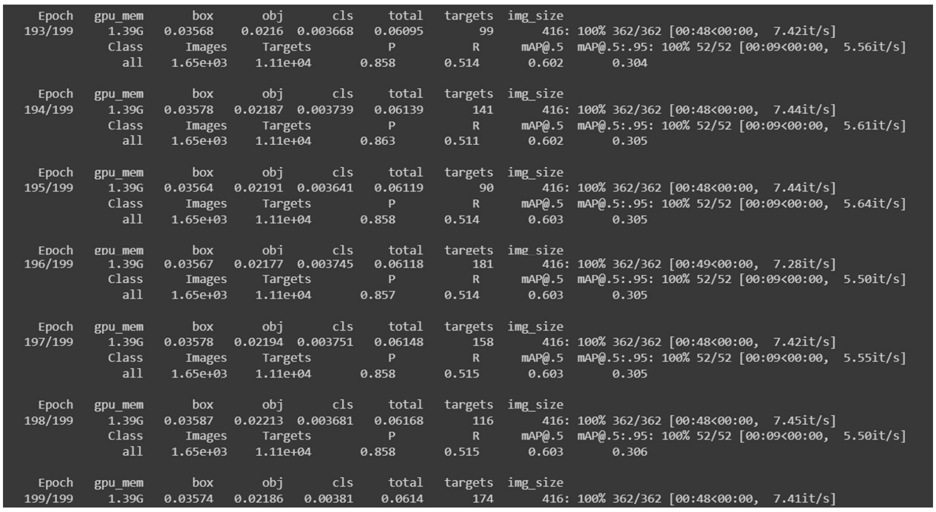
$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negatuve} \tag{2}$$

4.2. Model Training

Procedure for object detection using the YOLOv5 has being trained by our model to adopt the Udacity Self Driving Dataset and upgraded by Rob flow. Figure 5 presents the mode of operation of each epoch until it attains a total of 200 epochs.

Epochs are used to denote the number of successes that the deep learning algorithm has recorded through the aid of the training dataset. At the initial stages the object detection model was programmed to detect only 20 epochs, then over time it was then increase to up 200 epochs for optimum performance. In an interval of 3 hours 17 minutes, using python 3 google computer search engine backend, the computing powers provided by the Google collab is used to achieve 200 epochs. Figure 4 below presents models ground truth training data.

PERINTIS eJournal, **2024**, Vol. 14, No. 2, pp. 27-38



```
  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
193/199     1.39G   0.03568    0.0216  0.003668   0.06095        99       416: 100% 362/362 [00:48<00:00,  7.42it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.56it/s]
              all  1.65e+03  1.11e+04     0.858     0.514     0.602     0.304

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
194/199     1.39G   0.03578   0.02187  0.003739   0.06139       141       416: 100% 362/362 [00:48<00:00,  7.44it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.61it/s]
              all  1.65e+03  1.11e+04     0.863     0.511     0.602     0.305

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
195/199     1.39G   0.03564   0.02191  0.003641   0.06119        90       416: 100% 362/362 [00:48<00:00,  7.44it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.64it/s]
              all  1.65e+03  1.11e+04     0.858     0.514     0.603     0.305

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
196/199     1.39G   0.03567   0.02177  0.003745   0.06118       181       416: 100% 362/362 [00:49<00:00,  7.28it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.50it/s]
              all  1.65e+03  1.11e+04     0.857     0.514     0.603     0.305

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
197/199     1.39G   0.03578   0.02194  0.003751   0.06148       158       416: 100% 362/362 [00:48<00:00,  7.42it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.55it/s]
              all  1.65e+03  1.11e+04     0.858     0.515     0.603     0.305

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
198/199     1.39G   0.03587   0.02213  0.003681   0.06168       116       416: 100% 362/362 [00:48<00:00,  7.45it/s]
            Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 52/52 [00:09<00:00,  5.50it/s]
              all  1.65e+03  1.11e+04     0.858     0.515     0.603     0.306

  Epoch   gpu_mem       box       obj       cls     total   targets  img_size
199/199     1.39G   0.03574   0.02186   0.00381    0.0614       174       416: 100% 362/362 [00:48<00:00,  7.41it/s]
```

Figure 5: Processes run for 200 epochs

Figure 6. Ground truth training data

4.3. Model Evaluation

The Mean Average Precision (MAP) is usually measured as the first metric to realize the accuracy of a model. It adds up the average precision for a recall value starting from zero to one. Model Precision determines accuracy rate and exactness of predictions. The Recall procedure computes how best detection is conducted for the positives. Based on results generated, the MAP for Intersection over the Unions (IoU) threshold of 0.5 recorded was initially at 25.08% for 20 epochs training. Similarly for 200 epochs, the MAP valued at 60.36% presents that an enormous increase is realized based on our model's perfection.

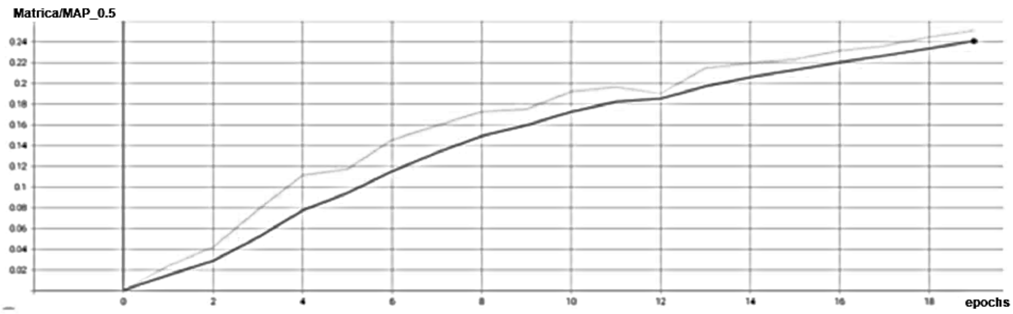


Figure 7. MAP for IoU of 0.5 for 20 epochs

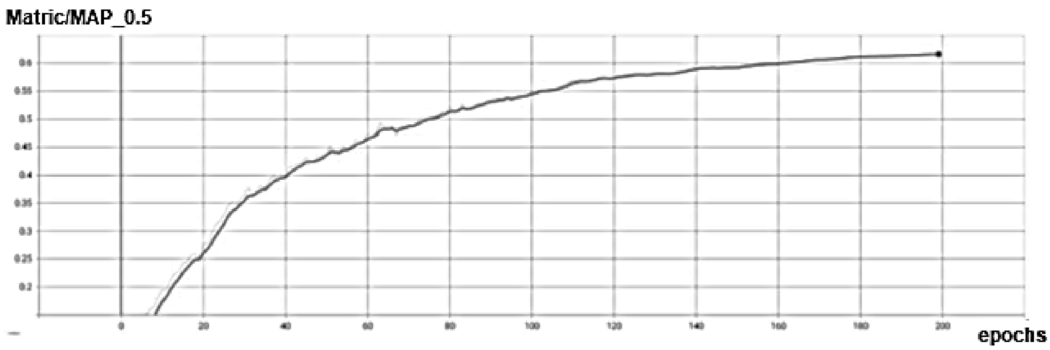


Figure 8. MAP for IoU of 0.5 for 200 epochs

The MAP for IoU threshold of 0.5 to 0.95 presents a value of 11.36% for 20 epochs. Similarly, a percentage of 30.64% was recorded for 200 epochs training. In both MAP graphs, it can be observed that the 200 epochs training presents a smoother and linearly interpolated graph. This means that an efficient performance of our model is realized.

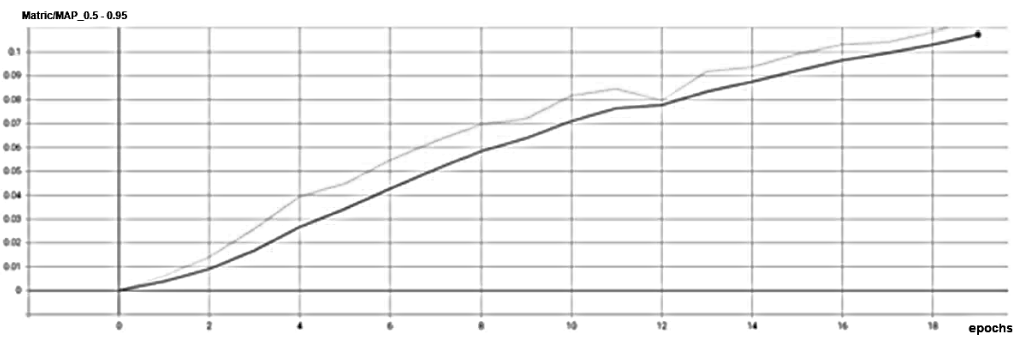


Figure 9: MAP for IoU of 0.5 to 0.95 for 20 epochs

PERINTIS eJournal, **2024**, Vol. 14, No. 2, pp. 27-38

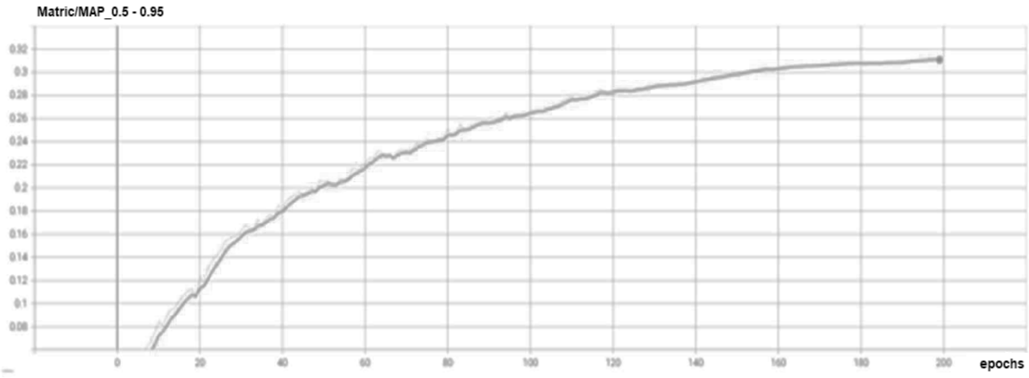


Figure 10: MAP for IoU of 0.5 to 0.95 for 200 epochs

Simulation evaluation using YOLO is conducted for real time datasets being encapsulated as presented in Figures 7 - 10. Based on the result, a biker is detected inform of a car. However, in some scenarios, a biker and a car cannot be differentiated. This model is purposely trained for successful car and bike detection and differentiation as a positively true. This enhances our result by presenting progress in the rate of accuracy of detecting objects in a self-driving situation.

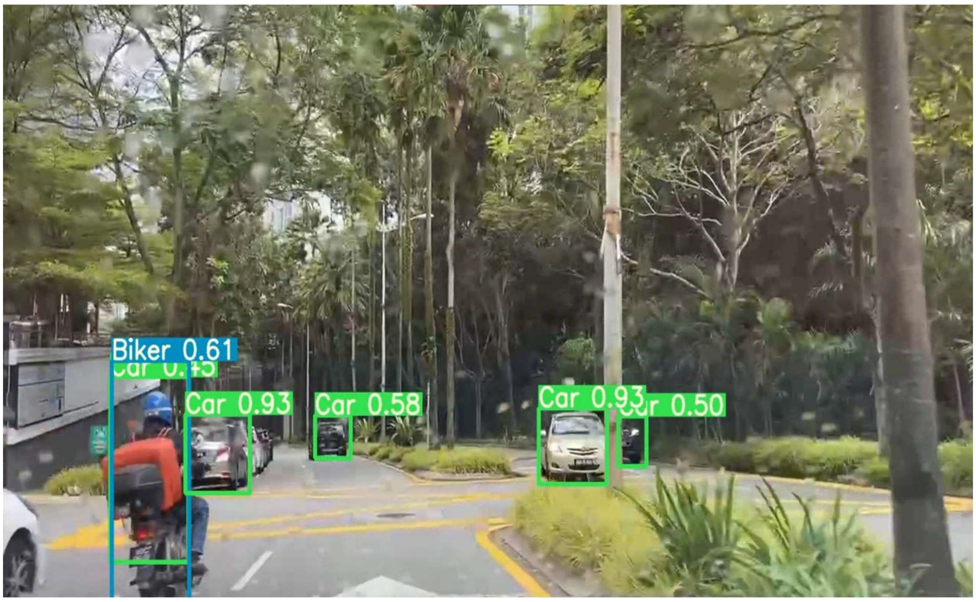


Figure 11: Inference on 20 epochs model

This form of detection presents less accuracy until more precision is added to the detection procedure for camera visibility in the data capture. Based on the information generated during model training, the result present enhanced precision as compared to other existing models. For MAP, recall value and lesser losses of box, object detection models are expected to be categorized for autonomous driving.

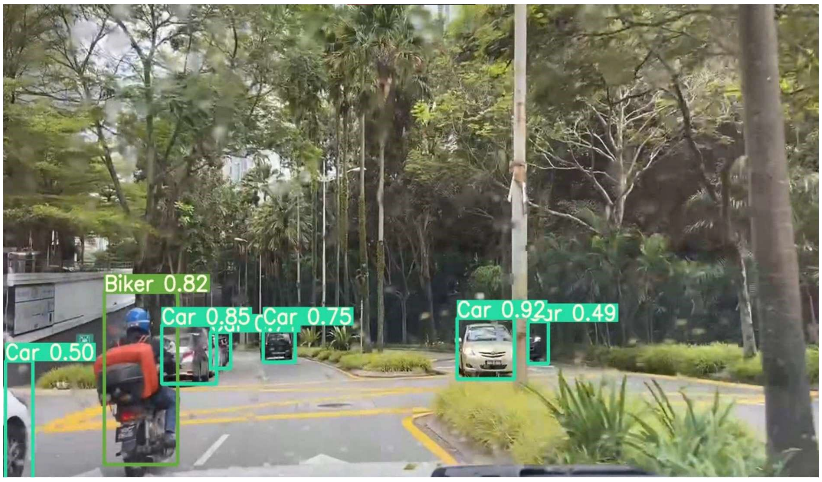


Figure 12: Inference on 200 epochs model

However, distance can affect the precision of evaluation performance in object detection models. Objects farther from the camera often appear smaller, less distinct, and harder to detect accurately, potentially leading to lower precision and recall. This is particularly relevant in autonomous vehicle applications, where detecting distant objects is crucial for safety. To address this, models can include data augmentation techniques that simulate objects at various distances during training. Additionally, loss functions can be designed to weigh bounding box predictions and classifications based on their spatial relevance or distance.

Typically, Google Colab provides access to GPUs like the NVIDIA Tesla K80, T4, P100, or A100, depending on the user's configuration and availability.

4.4. Comparison of Yolo Models

Our approach for vehicle detection procedure is compared with existing research using a trained and evaluated model. Our model being trained for autonomous driving using YOLO is presented as in table 1 below.

Table 1: Comparison for YOLO object detection model.

Model	Dataset	MAP	FPS
YOLOv2-608x608	COCO	48.1	40
YOLOv3-608	COCO	57.9	20
YOLOv3	Udacity	28.19	32
YOLOv4	Udacity	41.46	45
YOLOv5	Udacity	60.36	58

Based on comparison result of table 1 above, the YOLOv5 network survey presents better performance by generating Mean Average Precision as compared to the existing models. To ensure more precision of our model, future research is required to enhance the result generated during this simulation. Also, the FPS of the YOLOv5 object detection model is more efficient as compared to the existing approach. This also ensures the feasibility of YOLOv5 as the most effective tool to implement autonomous self-driving.

5.0 CONCLUSION

This paper implements an efficient approach of detecting moving vehicles in a smart city environment. This procedure is implemented using simulation with YOLOv5. An epoch parameter is measured in the vehicular network for precise measurement to generate an effective dataset. Results generated using YOLOv5 simulation are compared with existing approaches to determine the most precise model for easy implementation in a smart environment. The YOLOv5 network is analyzed with the Udacity dataset to determine model precision of up to about 61% precision with 50% recall at 58 frames per second. These values were validated with existing models for efficient validation and ease of real-life implementation. Although, some limitations were recorded in terms of GPU resources usage for Google Colab. These challenges interfered with some of our model training results during simulation runtime. A MAP of 60.36% is generally not sufficient for real-world autonomous driving use, especially in safety-critical environments. In the future, models will require generating customized data from the beginning of the simulation and to combine it with another dataset. This will ease data transmission and reception to enhance a model for better objects detections.

REFERENCES

[1] Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., & Ma, X. (2020). 3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. *Sustainable Cities and Society*, *54*, 102002.

[2] Ponnaganti, V., Moh, M., & Moh, T. (2020). Deep learning for lidar-based autonomous vehicles in smart cities. *Handbook of smart cities*, 1-25.

[3] Gupta, A., Anpalagan, A., Guan, L., & Khwaja, A. S. (2021). Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, *10*, 100057.

[4] Liu, Y., Yixuan, Y., & Liu, M. (2021). Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, *6*(2), 919-926.

PERINTIS eJournal, **2024**, Vol. 14, No. 2, pp. 27-38

[5] Pandey, A., Puri, M., & Varde, A. (2018, November). Object detection with neural models, deep learning and common sense to aid smart mobility. *IEEE 30th international conference on tools with artificial intelligence (ICTAI)*. 859-863.

[6] Nezhadalinaei, F., Zhang, L., Mahdizadeh, M., & Jamshidi, F. (2021, May). Motion object detection and tracking optimization in autonomous vehicles in specific range with optimized deep neural network. *7th international conference on web research (ICWR)*, pp. 53-63.

[7] Sanil, N., Rakesh, V., Mallapur, R., & Ahmed, M. R. (2020, January). Deep learning techniques for obstacle detection and avoidance in driverless cars. *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1-4.

[8] Mehajabin, N., Ma, Z., Wang, Y., Tohidypour, H. R., & Nasiopoulos, P. (2022, October). Real-time deep learning based road deterioration detection for smart cities. *18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 321-326.

[9] Pravallika, A., Hashmi, M. F., & Gupta, A. (2024). Deep Learning Frontiers in 3d Object Detection: A Comprehensive Review for Autonomous Driving. *IEEE Access*.