Section A-Research paper
ISSN 2063-5346

APPLICATIONS OF MACHINE LEARNING ON REAL TIME OBJECT DETECTION AND CLASSIFICATION IN AUTONOMOUS VEHICLES

RAMU V
Asst Professor in CSE (AI & ML) Department Balaji Institute of Technology & Science
(Autonomous) Narsampet, Warangal Urban District-506331
ramubits2022@gmail.com

HYMAVATHI SABBANI

Asst Professor in CSE (AI & ML) Department Balaji Institute of Technology & Science
(Autonomous) Narsampet, Warangal Urban District-506331
hymavathisabbanni@gmail.com

N LAVANYA
Asst Professor in CSE (AI & ML) Department Balaji Institute of Technology & Science
(Autonomous) Narsampet, Warangal Urban District-506331
lavanyamatcha@gmail.com

Rajesh Perugu
Asst Professor in CSE Department Balaji Institute of Technology & Science (Autonomous)
Narsampet, Warangal Urban District-506331
rajeshp526@gmail.com

ABSTRACT— Object detection is a kind of computer vision application that is now an essential component of a wide variety of consumer applications. Some examples of these applications include surveillance and security systems, mobile text recognition, and the diagnosis of disease using MRI or CT scans. Detection of obstacles is another essential part to enable autonomous driving. The ability of autonomous cars to have a reliable and secure driving performance is dependent on their perception of the world around them. This study introduces a YOLOv4-based, single-stage method for object detection that improves detection accuracy and enables real-time operation. It is crucial for the safe operation of an autonomous vehicle to be able to identify and follow nearby objects. Object detection, classification, and tracking algorithms are presented in this study. Everything is sorted into several categories based on whether it is in motion or not, and what it is (car, person, or something else). To detect and classify objects and estimate their position around the car, the proposed method combines data from a laser scanner with the state-of-the-art deep-learning network YOLO (You Only Look Once). Detecting obstacles is crucial for autonomous vehicles. Accurate and real-time detection of all roadside items is essential for the safe operation of high-speed vehicles. Recent years have seen a surge in interest in the question of how to strike a compromise between detection speed and precision. This work proposes a YOLOv4-based, single-stage approach for object detection that both increases detection precision and can run in real time.

Keywords: YOLO v3,v4, Object detection, autonomous vehicle machine learning.

1. INTRODUCTION

Object detection is a computer vision approach that identifies and localizes items inside an image. This may be accomplished by using an image. Object identification, in particular, involves drawing bounding boxes around the items that have been identified. This makes it possible to pinpoint the precise locations of the objects inside a picture. Object detection plays a significant part in surveillance and security, as well as in the monitoring of traffic and activities. It is used in the process of object tracking. For instance, following someone through a shopping mall. Object detection is essential for autonomous vehicles so that they can make decisions on how to proceed, such as whether to accelerate, stop, or turn. It is necessary for it to have knowledge of the locations of the things on the roadway. YOLO is used in the development of the suggested system for the purpose of object recognition and classification within picture data derived from traffic scenes.

When it comes to the deterioration of the environment, one issue that has captured a lot of people's attention is the fast acceleration of industrialization, urbanization, and the pace of population growth throughout the world. The world's population is growing at an alarming pace, which has led to a tremendous amount of environmental degradation, which has resulted in the state of the environment being in a terrible state. According to the findings of the study that was completed in 2019 and published in 2019, India generates more than 62 million tonnes (MT) of solid waste on an annual basis[1]. The world is now witnessing immensely severe repercussions on the economy, public health, and ecology, all of which may be ascribed to the pre-existing environmental issues and poor management of waste. These impacts are having a direct and immediate effect on the whole planet. As a result of this, the major focus has shifted towards the construction of smart cities throughout the world in order to offer effective and ecologically responsible waste management in metropolitan areas. In addition, the recycling of garbage not only makes it possible for research and development to take place, but it also integrates rubbish into successful business models that promote long-term, environmentally responsible expansion. Concerns, on the other hand, have been voiced about the need that garbage be separated according to whether or not it is biodegradable. Perception, planning, and control are the three subsystems that are often broken down into separate categories within autonomous driving systems [4].

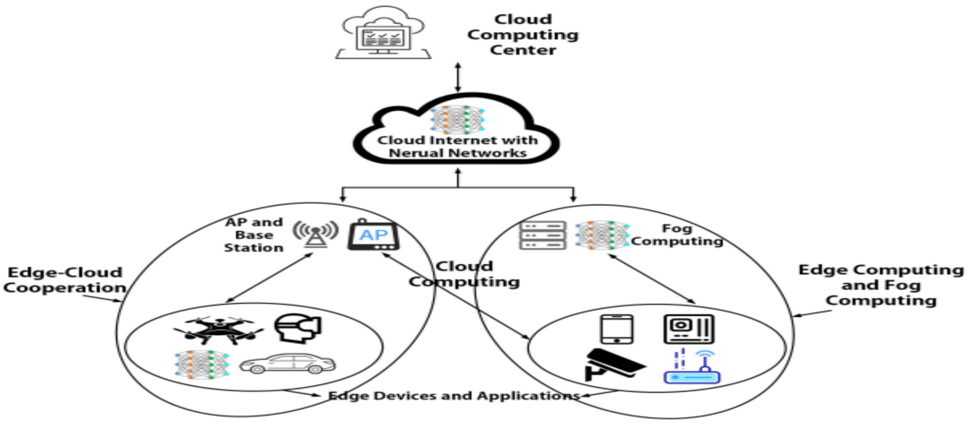

Cloud Computing Center
Cloud Internet with Nerual Networks
AP and Base Station
AP
Fog Computing
Cloud Computing
Edge-Cloud Cooperation
Edge Computing and Fog Computing
Edge Devices and Applications

Fig. 1. Structures for cloud and edge computing layers

We propose an artificial intelligence of things system here; we term it Edge YOLO. It works well with the existing ITS computer infrastructure. Using YOLOV4 as inspiration, we revamped the edge layer's network architecture and, after some light processing, came up with a more suitable solution we dubbed Edge YOLO. An edge computing device might have this algorithm preloaded upon it. In order to support a broad range of use cases for real-time object identification in ITS, as shown in Figure 1, we placed the neural network directly on the edge node. Moreover, we place the local device in charge of the collection and storage of picture data. There are three distinct neural network architectures for use in the cloud, on the edge, and in related applications as shown in figure 1. E-CC, Cloud Computing, edge Computing and Fog Computing.

The purpose of this paper is to analyze the perception issue and its effects on autonomous vehicle object detection in real time. A new era of autonomous driving may be on the horizon, but we are still far from realizing the full potential of this technology. Self-driving cars need to be able to adapt to and work in environments where many factors, such as humans, other cars, and animals, might change at any time. As a result, it is crucial to equip autonomous vehicles with reliable perception technologies that can aid and improve the vehicle's ability to comprehend the environment in which it is operating. When it comes to mapping their environments, self-driving vehicles have traditionally depended mostly on technology such as radar, lidar, global positioning system (GPS), and other sensors [4]. But, cameras are far more affordable than these sensors, therefore it would be very beneficial to use them more often into autonomous driving systems.

2. REVIEW OF LITERATURE

The study that was done by Lynch and colleagues mostly focused on the sensors that are essential for directing an AGV's navigation. They gave a concise summary presentation on the sensors that are used for AGV navigation. A study on the identification of obstacles in off-road environments utilizing a combination of sensors was provided by Hu and colleagues. It's possible that a single sensor has numerous limits and can't completely fulfill the needs of obstacle detection, but using many sensors may help you get around this obstacle detection problem. The authors have provided a concise summary of the most recent developments in fusion approaches and offered advice on how to choose sensors taking into account both their performance and their surroundings.

Guastella et al. presented an outstanding piece of work, in which they discussed new research that highlights learning-based approaches to resolving issues with environment perception. The information that the vehicle needs to comprehend its own position as well as its surrounds may be obtained via perception. They divided the learning-based technique into two major groups: those that were based on the terrain traversability and those that were based on the end-to-end method. The capacity to traverse terrain may be further broken down into categorization, regression, and a hybrid of the two approaches. They are primarily interested in addressing issues pertaining to driverless mobility in off-road settings.

Liu et al. solely used 3D lidar in their investigation of drivable ground, positive, and negative barriers, along with hanging obstructions. Throughout the course of their investigation, they largely identified barriers that stood in the way of discovering the territory that might be traveled. They conducted an analysis using 3D lidar points, focusing on the radial and transverse characteristics. After the detection of the impediments using these characteristics, the remaining areas are categorized as driveable zones for the AGV to go through.

Using lidar, Gao et al. also suggested a deep learning method. This technique took as input the vehicle's route and output a list of obstacles with their coordinates. The suggested network design takes the off-road environment's obscurity and uncertainty into account. Since it is done automatically, there is no need for humans to be involved in the process of labeling the barriers for their network. The fact that this network can process data with either no labeling at all or just a minimal amount of labeling and yet get good results is the primary benefit of using it.

In one unified model, Chen et al. identified road portions that could be traversed as well as impediments. They began by gathering data from lidar images and then transformed it into a lidar point cloud. The lidar point cloud was used to build a histogram map, which can now be seen from the vantage point of the car to show the portion of the road that is passable. In addition to the road that may be traveled, the barriers that surround it can also be seen.

3. RESEARCH METHODOLOGY

YOLO is different from the two-stage algorithm using region proposal to get regions of interest. Instead, it detects objects by segmenting the image into grid cells. Its output layer information includes bbox coordinates, confidence, and classification score. Therefore, it can detect multiple objects through a single stage, and the speed is much faster than two-stage algorithm. However, due to the fact that it predicts coordinates directly and not based on anchor, it is difficult to detect small objects. YOLOv2 adds BN layer after convolution layer, applies the idea of bbox based on anchor, multiscale training, and uses passthrough layer to fuse fine-grained features, which improves the accuracy compared with YOLO and YOLOv3; its backbone DarkNet53 applies residual connection to solve the problem of deep network gradient disappear; FPN feature fusion retains small object fine-grained features; multiscale prediction makes the network detect objects of different sizes. It has a more obvious improvement compared with YOLO and YOLOv2. The structure of YOLOv4 is shown in Figure 2.

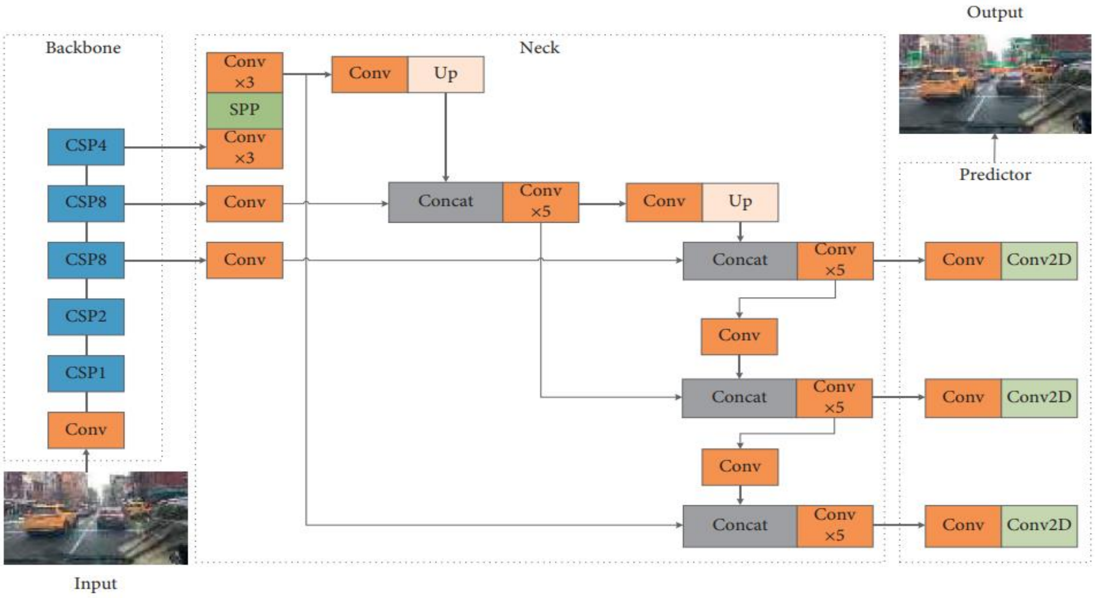


Figure 2: YOLOv4 structure

Our object detection network, Edge YOLO, follows this structure. Primarily, channel pruning in the network's backbone is performed to reduce the size of the stack compression model of the remaining block. The goal is to lessen the number of convolutional operations performed during the whole process of feature extraction. In YOLOV4, the bottom-up feature fusion is introduced using FPN since the original network PANet must repeatedly extract the feature layer. This results in a closed-loop network. We do this in the Neck area. Lightweight networks, which are better suited for mobile devices or devices without GPU, often overlook feature fusion and multi-scale prediction due to the fact that repeated extraction of features would result in a considerable degree of duplication. This is so because there will be a lot of duplication if features are extracted over and over again. We put a huge number of different feature fusion structures through their paces in the Neck layer, and in the end, the combination of SPP and FPN, along with the removal of the PANet network, was what was kept and put to use for the fusion of features. Upsampling and convolutional layers were used to link two neighboring networks that were of different sizes. Last but not least, with regard to the YOLO Head component, we have decided to stick with the original idea, which consists of a convolution combination of 3x3 and 1x1, followed by linearized output.

3.1 YOLOv3 Algorithm

One of the most well-known cutting-edge deep learning methods that allows for simultaneous object identification and categorization is YOLO (You Only Look Once). Previous methods used a pipeline execution architecture, which consists of many stages, to complete the object-detection job. Due to the pipeline's architecture and the necessity for separate training of each component, optimization progress is slow and difficult. Due to YOLO's ability to reduce object recognition to a single regression problem, these restrictions no longer apply. This does the forecasting of many bounding boxes and their respective class probabilities all at once. Training in YOLO is performed on entire pictures, unlike sliding window and area proposal-based systems, directly enhancing the detection performance. Nonetheless, YOLOv3's real-time speed, end-to-end training capabilities, high average accuracy, and generalisation capacity back up its effectiveness in performing complex object recognition tasks, including very microscopic objects

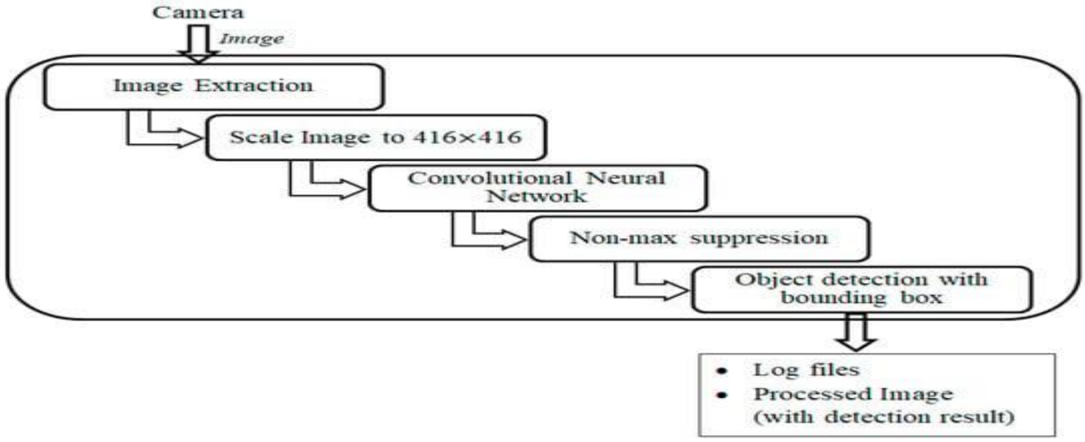

Camera
Image
Image Extraction
Scale Image to 416×416
Convolutional Neural Network
Non-max suppression
Object detection with bounding box
Log files
Processed Image (with detection result)

Figure 3. Outline of YOLOv3 algorithm.

The YOLOv3 approach (shown in Figure 3) just feeds an input image into a neural network, producing a vector of bounding boxes and class predictions. To feed the YOLOv3 neural network, a single picture is extracted and scaled down to 416416.416416. Using the darknet-53 framework, a YOLOv3 neural network was built, and its structure is shown in Figure 4. Layers of convolution, residuals, upsampling, and skip (shortcut) connections make up the network. YOLOv3's architecture is described in depth in a large body of published material.

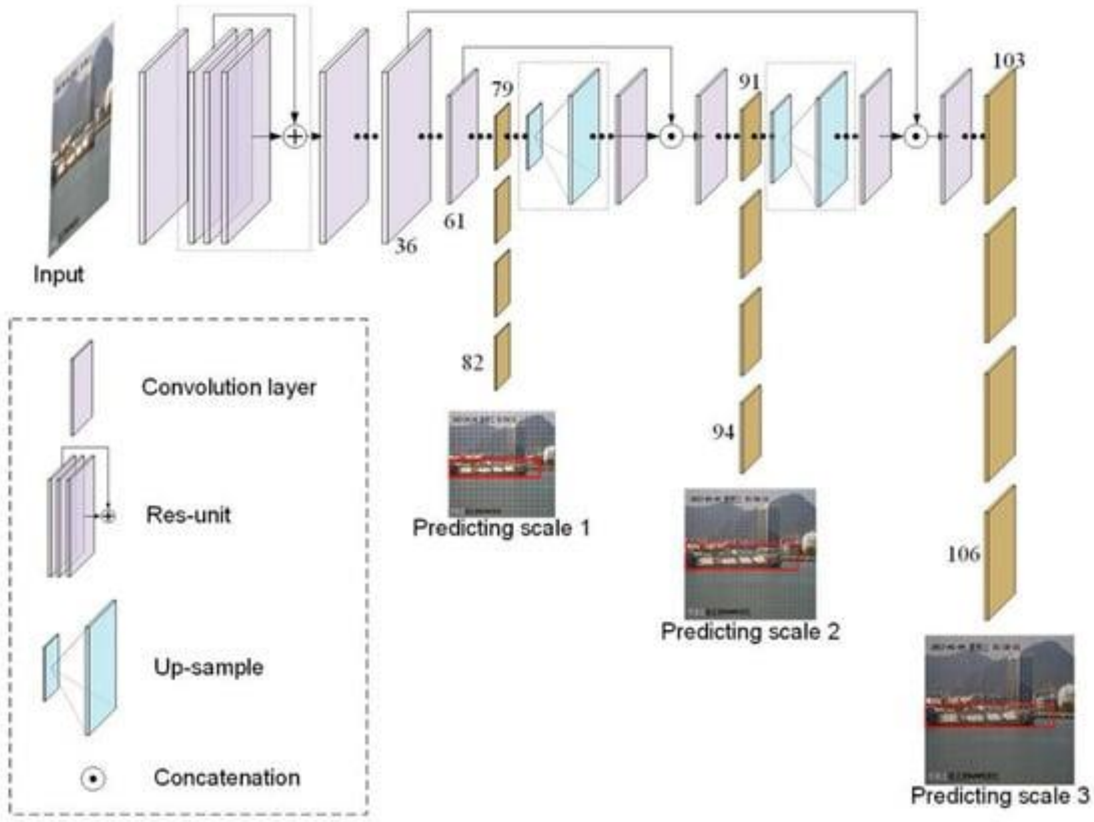


Figure 4. YOLOv3 architecture.

This article describes YOLOv3's performance in trash segregation by analyzing some of the essential important values examined all through the training process. The following are some examples of these essential ideals.

Precision

The level of precision is measured as a percentage of properly identified items relative to the total number of objects identified. Accuracy may be calculated mathematically as shown in Equation (1).

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

(1)

Recall

In order to measure how well an object detector performs in terms of recall, We compute the recall percentage by comparing the number of identified items to the number of ground truth objects. Equation (2) may be used to measure recall:

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \qquad \textbf{(2)}$$

where N_{TP} = True Positives, or the number of accurately identified items;

N_{FP} = False Positives, or the number of detected items that do not correlate to the ground truth;

N_{FN} = False Negatives, or the number of ground truth items that were not discovered.

Dataset Description

Beginning with the publicly accessible dataset COCO2017, which was derived from the Common Objects in Context (COCO) datasets that Microsoft funded and annotated in 2022, we do the following analysis. When it comes to object detection, the COCO competition is the gold standard. Annotations may be placed on eight different sorts of data, such as images and objects. Other image datasets are often used as benchmark data for object recognition, image segmentation, and image classification, however they generally include fewer targets per image.

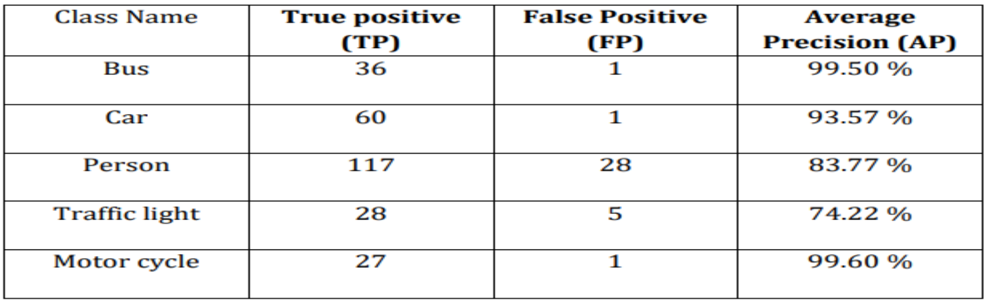
Yolo version3

In 2018, Yolo 3 was released, and it significantly outperformed the previous versions. Modifying the bounding boxes ever-so-slightly improved the predictions. To get the features and make the prediction, the version used 53 convolution layers. There was a time when the object detection model was applied to a picture on many scales and positions. When the image's score is high, it's likely that a detection of the target class item was made there. Version 3 of Yolo only uses a single neural network to process the whole picture. In comparison to classifier-based systems, it offers very little benefits. Unlike RCC, which needs thousands of evaluations for a single picture, this method simply needs a single assessment of the network. In comparison to RCC and even faster RCC, its speed is an astounding one hundred times faster. 2.7

Yolo version 4

While this edition was released in 2020, its original creator, Joseph Redmon, disavowed it for having numerous undesirable effects on society. Alexey Bochkovskiy, who also worked on the Original Windows version, created this variant. Several things may be identified instantaneously in a single picture. With its help, 9000 different types of courses may be anticipated. This Yolo version 4 is built on the CSPDarkent53 platform. This new framework is Yolo v4's foundation, and it has the power to improve CNN. The spine is PANet, while the head is Yolo v3. In the spine, spatial pooling is employed to improve perceptibility and zero down on the bare minimum of features needed.

4. RESULTS AND DISCUSSION



Class Name	True positive (TP)	False Positive (FP)	Average Precision (AP)
Bus	36	1	99.50 %
Car	60	1	93.57 %
Person	117	28	83.77 %
Traffic light	28	5	74.22 %
Motor cycle	27	1	99.60 %

Table 1 : Average Precision for each class

Table 1 shows the average precision of each class which is used to calculate the mean Average Precision.

As may be seen in Fig. 5, cars are recognized many times. With Non-Max Suppression, one may get a single detection of any item.

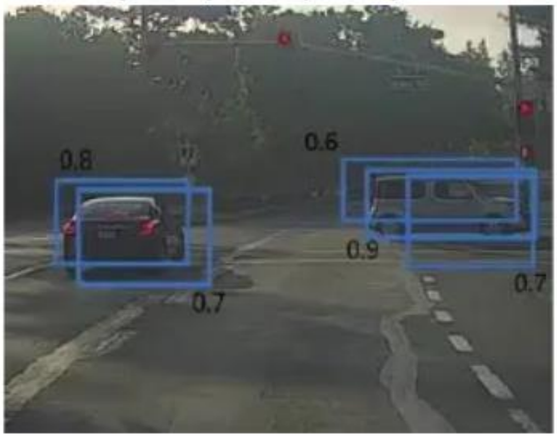
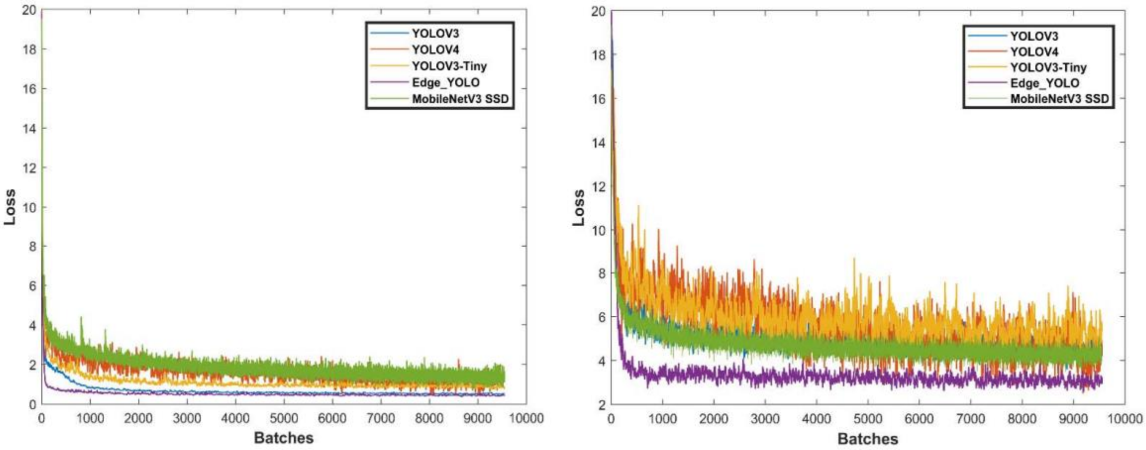


Fig. 5: Vehicles recognised more than once



(a) KITTI Dataset

(b) COCO2017 Datasets

Fig. 6. Training results for loss function by two Datasets.

Figure 6 demonstrates that the COCO2017 and COCO datasets were used to train the Edge YOLO network and four additional typical object identification networks. The YOLOV4 and YOLOV3 object identification networks, for example, have been trained for approximately 150,000 Batches to converge. The training of additional object identification networks has reached almost 100,000 Batches.

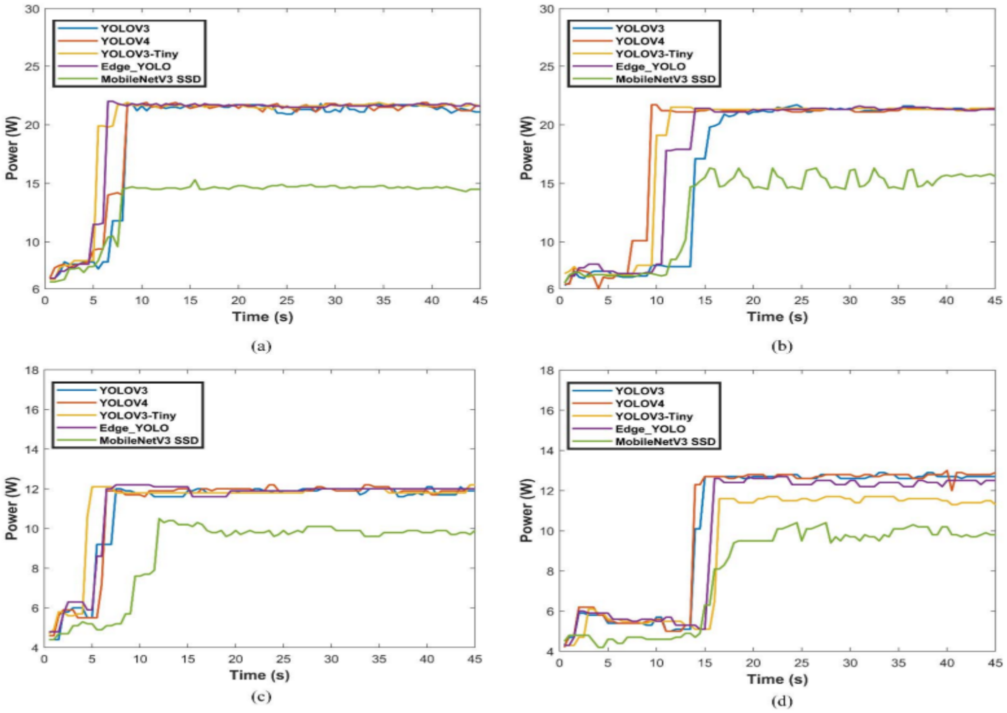


Fig. 7 The outcomes of Edge YOLO's interference after training on the KITTI and COCO2017 datasets.

Throughout the course of the functioning of the device that is attached to the vehicle, we make use of five distinct networks for object recognition and choose a portion of the total amount of energy used. We classify the model reasoning into four distinct categories, as follows: Figures 7a, 7b, 7c, and 7d depict COCO2017 datasets on Jetson Xavier, Nano, and Nano, respectively, from the 2017 conference.

Figure 8 compares the projected position of the RobotCar with its actual GPS position. Due to the lack of available ground truth for the RobotCar's location, results are evaluated in terms of how well they match up to actual measurements. A small amount of smoothing in relation to the observed location ensures that the predicted position closely tracks the measured position.

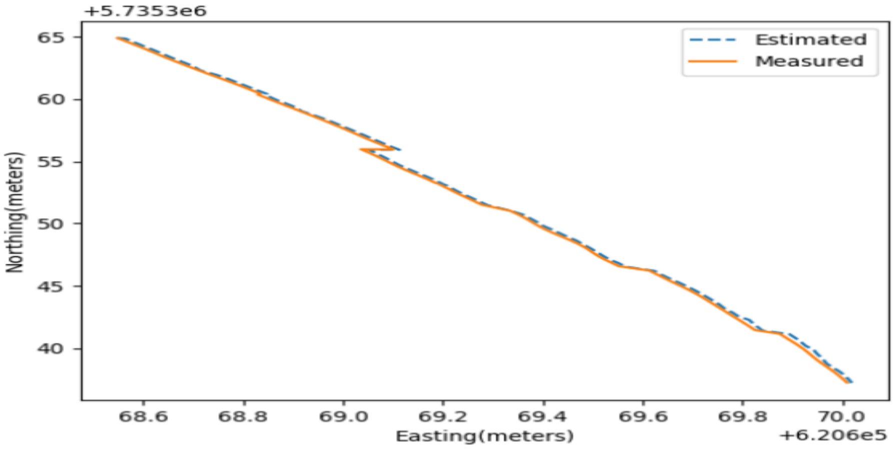
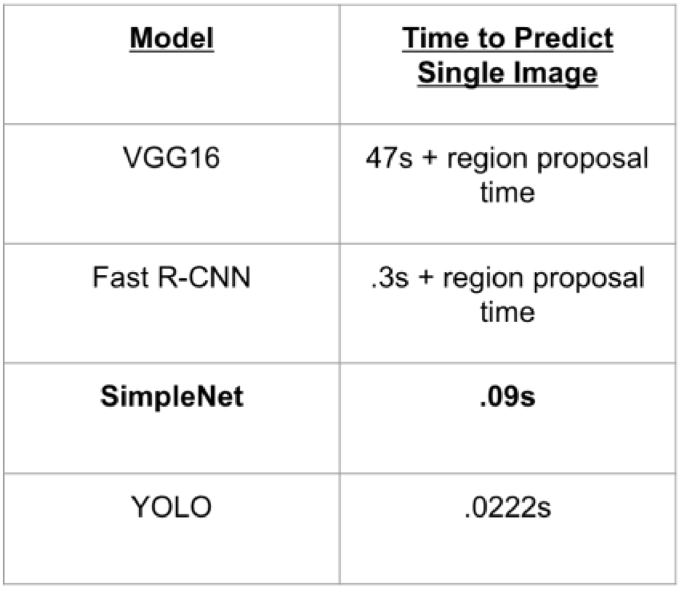


Figure 8: Estimation of postition of the Robotcar while tracking objects.

We also evaluate the prediction speed of our model against that of current work in the literature. At least 30 frames per second (or 0.033 seconds per image) of prediction accuracy is required for a detection method to be termed "realtime." Our model averages 0.092 seconds per image prediction time over 10 tests, which is three times slower than the bare minimum for real-time. Figure 9 shows the results when compared to other methods.



Model	Time to Predict Single Image
VGG16	47s + region proposal time
Fast R-CNN	.3s + region proposal time
SimpleNet	**.09s**
YOLO	.0222s

Figure 9: Prediction Speeds for Deep Object Detectors

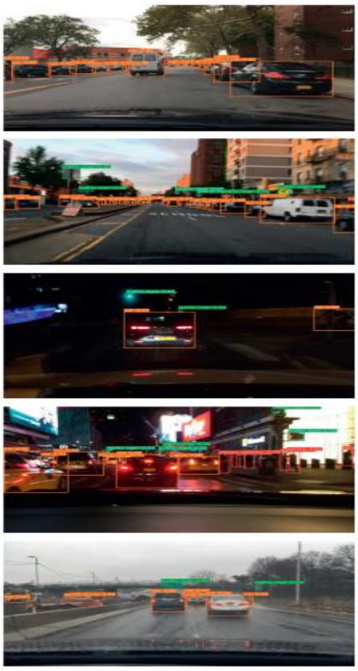


Figure 10: (a) YOLOv4 inference results.

Figure 10 is a graphical representation of the contrast between YOLOv4 and the new method. The third row reveals that, unlike YOLOv4, models 1 and 2 are able to spot traffic light objects in the dark.

CONCLUSION

The field of autonomous driving relies heavily on real-time object detection technology. Based on YOLOv4, this paper replaces SPP with RFB structure after backbone, integrates CSP structure with less computation into neck structure, and then adds CBAM and CA attention mechanism to make the neural network pay more attention to the object area containing important information, suppress irrelevant information, and improve detection accuracy, all in an effort to address the problem of insufficient accuracy of a one-stage detector in an autonomous driving scene. Edge For small-scale embedded devices with a single GPU, YOLO's entire network's 8 million parameters is ideal. This implementation of the YOLO series method on the edge layer is done so that it can strike a better balance between speed and accuracy. The typical power usage is also quite low, at just 16.8W. In addition, we hope to include edge YOLO into a wider variety of ITS frameworks and applications, as well as expand the number of edge computing devices on which it is deployed. Working on Darknet required the employment of algorithms, namely Yolo version 3 and Yolo version 4. Both of these models provide results for still photographs, moving movies, and webcam footage captured in real time. According to what we have seen, both of the algorithms are quick. When it comes to making accurate predictions, Yolo version 4 has shown to be superior when using the same dataset. As the dataset photos were taken while the automobile was moving, the model may be deployed in a moving vehicle to identify things in the same manner that they were captured.

REFERENCES

1. Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: Proceedings of the IEEE International Conference on Computer Vision. Vol. 2015 Inter. 2015, pp. 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123. arXiv: 1502.01852.
2. Zhong-Qiu Zhao et al. Object Detection with Deep Learning: A Review. Tech. rep. 2018. arXiv: 1807.05511v1.
3. Joel Janai et al. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art”. In: (2017). ISSN: 10495258. DOI: 10.1007/1-4020-3858- 8_15. arXiv: 1704.05519.
4. Scott Pendleton et al. “Perception, Planning, Control, and Coordination for Autonomous Vehicles”. In: Machines 5.1 (2017), p. 6. ISSN: 2075-1702. DOI: 10. 3390/machines5010006. URL: http://www.mdpi.com/2075-1702/5/1/6.
5. Lynch L., Newe T., Clifford J., Coleman J., Walsh J., Toal D. Automated Ground Vehicle (AGV) and Sensor Technologies- A Review; Proceedings of the 2019 13th International Conference on Sensing Technology (ICST 2019); Sydney, Australia. 2–4 December 2019; pp. 347–352. [CrossRef] [Google Scholar]
6. Hu J.-W., Zheng B.-Y., Wang C., Zhao C.-H., Hou X.-L., Pan Q., Xu Z. A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments. *Front. Inf. Technol. Electron. Eng.* 2020;**21**:675–692.

doi: 10.1631/FITEE.1900518. [CrossRef] [Google Scholar]

7. Guastella D.C., Muscato G. Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured Environments: A Review. *Sensors.* 2020;**21**:73. doi: 10.3390/s21010073. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
8. Liu T., Liu D., Yang Y., Chen Z. Lidar-based Traversable Region Detection in Off-road Environment; Proceedings of the 38th Chinese Control Conference (CCC2019); Guangzhou, China. 27–30 July 2019; pp. 4548–4553. [CrossRef] [Google Scholar]
9. Gao B., Xu A., Pan Y., Zhao X., Yao W., Zhao H. Off-Road Drivable Area Extraction Using 3D LiDAR Data; Proceedings of the Intelligent Vehicles Symposium (IV); Paris, France. 9–12 June 2019; pp. 1505–1511. [CrossRef] [Google Scholar]
10. Chen L., Yang J., Kong H. Lidar-histogram for fast road and obstacle detection; Proceedings of the IEEE International Conference on Robotics and Automation (ICRA); Singapore. 29 May–3 June 2019; pp. 1343–1348. [CrossRef] [Google Scholar]
11. Katramados I., Crumpler S., Breckon T.P. *Lecture Notes in Computer Science.* Volume 5815. Springer; Berlin/Heidelberg, Germany: 2009. Real-Time Traversable Surface Detection by Colour Space Fusion and Temporal Analysis; pp. 265–274. [CrossRef] [Google Scholar]
12. Shaban A., Meng X., Lee J., Boots B., Fox D. Semantic terrain classification for off-road autonomous driving; Proceedings of the Machine Learning Research (PMLR); Almería, Spain. 5–7 October 2022; pp. 619–629. [Google Scholar]
13. Gao B., Hu S., Zhao X., Zhao H. Fine-Grained Off-Road Semantic Segmentation and Mapping via Contrastive Learning; Proceedings of the IRC 2021: IEEE International Conference on Robotic Computing; Taichung, Taiwan. 15–17 November 2021; pp. 5950–5957.
14. Joseph Redmon, Ali Farhadi “YOLOv3: An Incremental Improvement” University of Washington: arXiv:1804.02767. Available at University of Washington (https://pjreddie.com/media/files/papers/YOLOv3.pdf)
15. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, (April 23 2004)” YOLOv4: Optimal Speed and Accuracy of Object Detection”: arXiv:2004.10934 [cs.CV]