International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Small Object Detection in Autonomous Cars Using a Deep Learning

Adarsh Chaturvedee[1], Raghad Al-Shabandar[2], Ammar H. Mohammed[3]
[1]Independent Researcher, India
[2]Consultant of Artificial Intelligence Applications, Iraqi Prime Minister Office, Iraq
[1]adarsh0602@hotmail.com, [2]alshabnadrraghad@gmail.com, [3]ammarhussen659@gmail.com

Corresponding author email: adarsh0602@hotmail.com

***Abstract*—** In computer vision, object detection plays a vital role in ensuring the safety of a self-driving car. The most notable example of this continuing exploration and enhancement in the field is the Google Self-Driving Car Project, currently known as Waymo. Although many prior studies have utilised several object detection models to improve the efficiency of autonomous cars, each comes with its own set of challenges. The major challenge in the development of self-driving cars is latency. The latency refers to the delay between the processing of input data captured from the camera and the decision taken by the machine learning algorithm to move and direct the car on a safe road. To address these issues, we propose a MobileNet SSD framework by hyperlinking MobileNet with SSD, making it sufficient for real-time applications. The model utilises two types of sparable convolutions, namely spatial separable convolutions and stepwise sparable convolutions. The result demonstrates the efficiency of our proposed MobileNet SSD model in reducing computational costs and decreasing latency.

***Keywords*— Computer Vision, Convolutional Neural Networks, Google Self-Driving Car Project, MobileNet SSD, Machine Learning**

1. Introduction

Object detection is one of numerous stratifications of computer vision and artificial intelligence. There has been a lot of emphasis on developing capabilities such that self-driving vehicles can become a reality in the modern world[1]. The development in Artificial Intelligence and computer vision are keys to making autonomous vehicles reach a level of efficiency, competence and safety that they become more widely used in the world [2].

Machine learning is a promising tool that has been applied to detect objects in self-driving cars. A large amount of data can be collected from cameras to feed into machine learning model[3] .In order to recognize specific entities in an image or frames using bounding boxes, or anchor boxes in deep learning parlance, the machine learning model should be train in a good amount of data [2].

Speed object detection is a very important aspect that should be considered when selecting the machine learnings model, the model should be able to detect small objects on real time. High-speed object detection is the ability of algorithms to detect objects and obstacles in their surroundings[4].

With high-speed object detection, ensuring the safety of passengers, other road users, and the vehicle itself becomes achievable. It can also help autonomous vehicles to navigate through different weather conditions such as heavy rain, snow, and fog, which can obscure a human driver's vision[5].

Although Machine learning has been successfully applied to detect objects in autonomous vehicles, the object detection is facing main challenges precision and latency [4] [6]. precision is what in object detection might take a hit a high speed with proper localization, the autonomous system will work well[6].

Latency refers to the time takes to detect objects; it holds a slightly higher priority over detecting which type of object is being detected precisely. The latency is the main factor in achieving the optimal speed object detection. This task becomes even more important if the objects are smaller because the vehicle doesn't need to know what it is, it simply needs to know where it is and avoid touching it[7]. To the best of our knowledge, detection time for small objects with a marginal loss of precision has not been done yet.

The contribution of the paper are as follows (1) Various deep models are utilized to detect small objects in images and videos with reasonable precision whilst reducing latency. (2) MobileNet SSD framework is constructed. This framework utilizes depth-wise separable convolution and spatial separable convolutions. This highlights the capability of our proposed MobileNet SSD framework in enhancing the computation speed and reducing the latency.

The remainder of this paper is organized as follows; section 2 Background provides an overview about self-driving followed by related work. Section 3 give the brief description about the deep learning in object detection. The section 4 presents the detailed of methodology, which includes 1 database description alongside with analysis and results of two experiments Single Image Outputs. The conclusion and future works are described in section 5.

2. Background

Self-driving cars, also known as autonomous vehicles, are cars that can operate on the road without human intervention. Self-driving cars are designed to operate without human input by using various sensors, cameras, and algorithms that process data and make decisions. These vehicles have the potential to revolutionize the transportation industry, making travel safer, more efficient, and less expensive. However, the development of self-driving cars has been a challenging process, and there are still many hurdles to overcome[8].

The idea of self-driving cars has been around for more than a century. In the 1920s, engineers developed the first automatic transmission, which allowed drivers to shift gears without

using a clutch pedal. In the 1950s, the first cruise control system was introduced, which allowed drivers to maintain a consistent speed without pressing the accelerator pedal. In the 1980s, researchers began developing autonomous vehicles, and in 1987, the first self-driving car was developed in Germany[8], [9].In the early 2000s, several car manufacturers began experimenting with self-driving cars. In 2004, the Defense Advanced Research Projects Agen.

In recent years, several companies have been working on developing self-driving cars, including Google, Uber, Tesla, and General Motors. These companies have invested billions of dollars in research and development, and self-driving cars are expected to become a common sight on roads in the near future[1].

Self-driving cars are still relatively expensive, and the cost of developing and producing these vehicles can be prohibitive. This has limited the availability of self-driving cars and has slowed the pace of development in the field. The elaborate autonomous system which includes a host of sensors, programs and actuators themselves can add an overhead to the cost of the car making it less feasible to buy than a manually operated one.

The major challenges in the development of self-driving cars include Safety: Safety is one of the biggest concerns when it comes to self-driving cars. Autonomous vehicles must be able to operate safely on the road, even in unpredictable situations[10].

Liability is another concern when it comes to self-driving cars. In the event of an accident involving a self-driving car, it can be difficult to determine who is responsible for the accident.

Artificial intelligence has been successfully applied to control and regulate safety and liability in self-driving. Various machine learning algorithms have been implemented to detect and locate objects with images and videos[10], [11].

Object detection in autonomous cars plays a vital role in detecting and recognizing objects and their surroundings. Thus,

The self-driving car could locate their surroundings precisely, make accurate decisions and ensure the safety of the car.

2.1 Literature Review

Computer vision have gained increased attention by researchers in self-driving car. You Only Look Once (YOLO), Single Shot Multibox Detectors (SSD), Region-Based Convolutional Neural Networks (RCNN), and Faster-RCNN. Research has been conducted in recent years to assess the performance with the efficiency of object identification algorithms in specific cases and environments. Although, research has barely kept up with improved object identification algorithms like YOLOv5 and YOLOv7. YOLOv3 has been the standard algorithm in the object detection space. Having the most promising results in detecting vehicles and persons which are two of the most common found entities on roads, makes it that much more popular.

The author in reference [12] was employed convolutional neural network to detect the object for border tree trunks of urban roads. The result illustrate that deep learning performs well in respect to high precision with low detection time, but it was trained on a comparatively small dataset, and training on a larger set of data to run the model on multiple frames or videos is suspected to take a lot of time[12].

The author in [13] mentioned potholes are an important factor to consider in self-driving cars, The detection process involves a combination of sensors and cameras to detect the damage road areas. The author found that YOLOv3 is highly efficient since algorithm capable to efficiently identify potholes. and the result reveals that it outperforms SSD, HOG and Faster-RCNN.

According to related work, YOLOv3 is identified as the best object identification method. It is not just to identify items on the street, but also to detect objects in completely disjointed domains such as face mask detection as well. YOLOv3 outperforms Faster-RCNN in terms of performance and inference time[14].

Although Faster-RCNN requires longer training time, it is more accurate. Many studies have recently been conducted to compare different iterations of the YOLO object identification method. On YOLOv3 and Faster-RCNN, comparative research on object detection that frequently occurs in traffic settings has been conducted[15][16]. The result shows that Faster-RCNN, gains the promising result, However, it cannot be implemented efficiently in a real-world scenario due to, the latency could be very slow [15].

3. Machine Learning

This section highlights the current machine learning for object detection .A review of the relevant work in the ER domain is also explored. We highlight the importance of active learning and semi-supervised machine learning in labelling data.

3.1 Deep Learning in object Detection

Deep learning technology has been utilized to recognize and identify objects in self-driving cars in recent years, influenced by the rapid advancement of deep learning technology in computer vision tasks[17].

Convolutional Neural Networks (CNN) are a deep learning algorithm that is widely used for image classification and object detection tasks. They are designed to automatically learn and extract meaningful features from images, such as edges, shapes, and textures, by processing the images through a series of convolutional and pooling layers. CNNs have achieved state-of-the- art performance on a variety of computer vision tasks, including image classification, object detection, semantic segmentation, and image captioning. They have been widely used in a variety of industries, including self-driving cars, medical image analysis, and security and surveillance systems.

Region-based Convolutional Neural Networks (R-CNN) are an extension of CNNs that were designed to improve the efficiency and accuracy of object detection tasks. R-CNNs work by first generating region proposals, which are potential object locations in an image. Then, they use a CNN to extract

features from each region proposal and classify them into different object categories. R-CNNs were a major breakthrough in object detection because they achieved significantly higher accuracy than previous methods, while also being able to detect multiple objects in an image. However, R-CNNs were computationally expensive and slow to train and test, making them impractical for real-time applications.

Faster R-CNN is an extension of R-CNN that addresses its limitations by introducing a Region Proposal Network (RPN) that generates region proposals directly from the feature map of the CNN. The RPN shares convolutional layers with the CNN and is trained end-to-end with the CNN, resulting in a single, unified network that can generate region proposals and perform object detection simultaneously. This allows Faster R-CNN to achieve higher accuracy and faster inference speeds than R-CNN, making it practical for real-time object detection applications. Faster R-CNN has become a popular method for object detection and has been used in a variety of industries, including autonomous driving, robotics, and surveillance.

YOLO (You Only Look Once) is another popular object detection algorithm that is similar to R-CNN and Faster R-CNN. YOLO uses a single neural network that takes an entire image as input and outputs a set of bounding boxes and class probabilities directly. This means that YOLO performs both region proposal and classification in a single forward pass, making it much faster than R-CNN and Faster R-CNN. YOLO also has the advantage of being able to detect objects at different scales and aspect ratios, and it has achieved state-of-the-art performance on several object detection benchmarks.

YOLO-based R-CNN is a hybrid approach that combines the strengths of both YOLO and R- CNN. This approach has the potential to achieve higher accuracy than YOLO alone, while still maintaining the speed advantage of YOLO

YOLO works by dividing the input image into grid cells and using each grid cell to forecast the existence and placement of items within the cell. Because it does not require an external region proposal method, this approach is substantially faster than R-CNN and its derivatives.

Single Shot Detectors (SSDs) are a class of object detection algorithms that, like YOLO, take an entire image as input and generate a set of bounding boxes and class probabilities directly, without the need for explicit region proposal. The main advantage of SSDs is their speed, which is achieved through the use of a single convolutional network that simultaneously performs both object localization and classification. This means that SSDs can achieve real-time performance even on low-powered devices, such as smartphones and drones.

SSDs achieve high accuracy by using a multi-scale feature map, where different layers of the network are responsible for detecting objects at different scales. This allows SSDs to detect objects of different sizes and aspect ratios, while still maintaining the speed advantage of a single-shot approach. SSDs have been shown to achieve state-of-the-art performance on several object detection benchmarks, including COCO and PASCAL VOC. They have been widely used in a variety of applications, including robotics, self-driving cars, and video surveillance systems, where real-time performance is critical.

The incorporation of the anchor notion from Faster R-CNN is one of the fundamental aspects of SSD. In each feature map, the SSD algorithm generates a default box with variable ratios and sizes and then uses the model-generated coordinates and class values to create the final bounding box. This enables predictions for objects of varying sizes, enhancing performance and speed by replacing the Fully Connected Layer.

4. Methodology

In this paper, multiple object detection algorithms are employed with the aim to detect small objects in images and videos with reasonable precision whilst reducing latency.

Three distinct deep learning models are used in this study these are the YOLO model, SSD, and Faster RCNN. The You Only Look Once (YOLO) model is a real-time object detection that uses a single feedforward pass to process images[18][19].

We use, the YOLO V3 is a real-time object detection model that uses the features learned by a deep convolutional neural network [20].The model employs 75 convolutional layers, as well as up-sampling layers and skip connections, to create a single neural network that acts on the full image. One of the most significant features of YOLO V3 is its capacity to recognize items at numerous scales; however, this increase in accuracy comes at the expense of slower processing speed and a diminished ability to detect small objects that appear in group [20].

The Faster R-CNN (Region-based Convolutional Neural Network) is an object identification system that generates object suggestions using a region proposal network (R1PN). The R-CNN and its derivatives utilize a two-stage detection approach, with a region proposal network identifying potential objects in the image, followed by a convolutional neural network (CNN) performing object detection on the identified regions. This method has been shown to produce accurate results, but it can be computationally expensive due to the need for two separate models[21].

The Single Shot Detector (SSD) is another real-time object detection system that detects objects at various scales by employing a multi-scale feature extraction network[22].

SSDs, use a single-shot approach that performs both region proposal and object detection in a single step. This method utilizes anchor boxes like YOLO, allowing for more accurate detection of objects with different aspect ratios and scales. Additionally, SSDs can handle larger input sizes, leading to more precise object detection.

The SSD has limits in recognizing small objects because it does not take into account the context outside of the detection boxes. To address this issue, an enhanced SSD technique was considered. We merged the MobileNet architecture and the Single Shot Detector (SSD) framework. The MobileNet is

computer vision model allow to deploy model on resource-constrained devices like smartphones [23].
The framework MobileNet SSD uses depth-wise separable convolution and spatial separable convolutions. resulting in improved accuracy and efficiency. The depth-wise separable convolution maps each input channel to its matching output channel individually, while the spatial separable convolution convolves along the x and y axes. In this approach width multiplier is implemented in the depth-wise separable convolution and a resolution multiplier in the spatial separable convolution, leading to a significant reduction in the number of operations required for the method and the computational cost[24].
Various metrics are used to evaluate the algorithm's performance, including mean Average Precision (mAP), Precision, Recall, F1, these metrics are critical in determining the algorithm's accuracy by specifying the bounding box coordinates of objects.

4.1 Database Description

The Microsoft Common Objects in Context (COCO) dataset has been used in this study, the COCO is a large-scale benchmark for object detection, segmentation, and captioning images[25]. It was released in 2014 by Microsoft Research, and it has since become one of the most widely used datasets for evaluating computer vision algorithms. The dataset consists of over 330,000 images, each with multiple object annotations, and it covers a wide range of object categories, including people, animals, vehicles, and household items.
The COCO dataset is notable for its high-quality annotations, which have been manually verified by human annotators[26]. The image annotation can define the process of labeling an image in database.
The annotations include the bounding box coordinates of each object, as well as its category label, segmentation mask, and key point locations (in the case of human poses). The dataset also includes captions for each image, which describe the objects and their relationships in natural language. The annotations in the COCO dataset are highly accurate, with an average IoU (Intersection over Union) of 0.76 for object detection and an average precision of 0.56 for instance segmentation [25] [26].More details about database can be found in Table (1).

4.2 Experiment Setup

Two sets of experiments have been conducted in this paper. In the first experiment, the output was generated from a single image frame while in the second experiment, the output was driven from videos. We employed three object detection models these are YOLO, SSD, and R-CNN. Many pre-trained model files and configuration files, including YOLOv2.weight, YOLOv2.cfg, YOLOv3.weight, YOLOv3.cfg, SSD deploy. caffemodel, and SSD deploy. prototxt has been downloaded in order to train these models.
Additionally, several Python 3.7 packages are used such as NumPy, Imutils, OpenCV, and PyCharm are. The OpenCV is a popular library for computer vision applications such as image and video processing.
In the first step, we capture the images from COCO database that was released in 2015 where 83,000 images are allocated for the training set 41,000 images used as validation photos, and 41,000 are test photos. The 81,000-picture extra test set is also selected that was released in 2015 contained all the earlier test photographs as well as 40,000 more.
In the second step, we feed these images to deep-learning model-generated outputs. More details about the experiments can be found in the next section.

Table 1 : COCO Annotations -Features

Annotations -Feature	Description
id	It represents annotation's unique identifier
image _id	It represents image's unique identification
Category_ id	It represents the category id of the object in the image [1: 80] categories
Category_ name	
segmentation	It represents the item's segmentation mask
area	It represents the area of the object in the image
bbox	It represents the object's bounding box coordinates
iscrowd	
Category_ Name	It represents name of each object's

4.2.1 Experiment 1: Single Image Outputs

In the first experiment, generating output for single image is considered, a wide variety of images from the MS-COCO dataset were fed into three machine learning models. The analysis is shown side by side so that a comparison has been made between the YOLO and MobileNet SSD.
In term of self-driving car, we should explore how algorithms perform in outdoors , the figure (1) shows that images belong to the outdoors, especially those involving roads and vehicles, the MobileNet SSD is surprisingly efficient in detecting the objects that are present in the image in comparison to the more seasoned YOLO algorithm.In the image above, MobileNet SSD is able to detect all the objects that YOLO detected which is a big step in matching the YOLO algorithm in terms of accuracy.
With regards to moving objects the figure (2-a) YOLO (left) detects more moving objects on the street than MobileNet SSD (right) which could be detrimental for the safety of passengers in the vehicle. MobileNet SSD can identify the vehicles that are right in front of the camera, but even a slightly obscured car to the left remained untagged by MobileNet SSD. This is where YOLO shows that it detects more objects, more accurately than SSD.
Another aspect to consider for this image is that MobileNet SSD detects the lamp post as a "boat" in the image. This could

be due to the faster recognition of SSD as compared to YOLO.

We can see in figure (2-b) moving objects detection consider Faster R-CNN algorithm, the figure shows a lesser number of objects detected with the Faster R-CNN algorithm when compared to YOLO. However, the details captured by the model is slightly better than YOLO as it identifies wheels, slightly obscured cars and some of the traffic on the other side of the street. This was something that YOLO could not efficiently do, and picked up on all the details.

Figure (3-a) also shows Moving vehicles image when there are more obstacles, the finding shows that objects can be detected by the MobileNet SSD algorithms very well. YOLO was unsuccessful in detecting the motorcyclist towards the left of the image. It also detected lesser objects than MobileNet SSD. As mentioned, MobileNet SSD detects more objects on the street and does not miss out on small objects such as the motorcyclist as well. It is able to distinguish between the motorcyclist on the left and the motorcycle itself.

Figure(3-b) demonstrate that Faster R-CNN has a comparable precision to the YOLO model where the identification of details such as recognition of wheels is an additional feature than Faster R-CNN was able to pick up.

However, it was unable to pick up the information about the motorcycle to the left of the image. This was something which the algorithm did exceedingly well and was even able to distinguish between the ride and the motorcycle. An amount of taken time was computed for same set of above images for each algorithm.

The result shows the MobileNet SSD algorithm took 0.025 seconds to finish the object detection process. That is approximately 16 times faster than the YOLO algorithm.

The Faster R-CNN model took 0.14 seconds to detect a frame and form 100 bounding boxes. This is approximately 2.75 times faster than the YOLO algorithm. However, since the MobileNet SSD algorithm detects objects within 0.025 seconds, MobileNet SSD are approximately 6 times faster than Faster R-CNN models for object detection. More details can be found in table 2

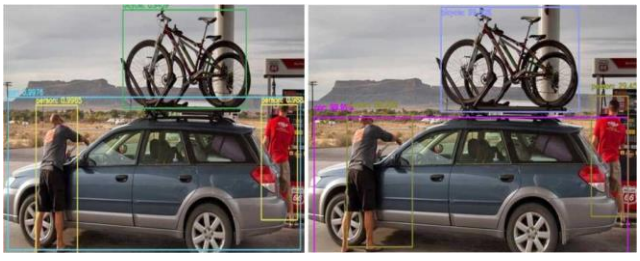

Figure (1-a): vehicles image as detected by YOLO (left) and SSD (right)

Figure (2-a) a: Moving vehicles image as detected by YOLO (left) and SSD (right)

Figure (2-b) a: Moving vehicles image as detected by Faster R-CNN

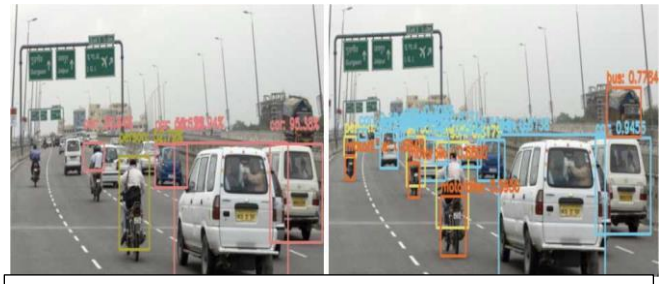

Figure (3-a): Moving vehicles image as detected by YOLO (left) and SSD (right)

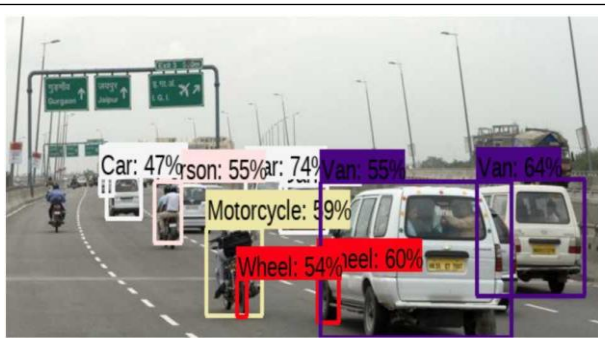

Figure (3-b): Moving vehicles image as detected by R-CNN

Table 2: Time based comparison of YOLO, Faster R-MobileNet SSD for Experiment 1

Parameter	YOLO	Faster R-CNN	MobileNet SSD
Speed	Slow	~3x faster than YOLO	~16x faster than YOLO)
Time	0.4 seconds/frame	0.1 seconds/frame	0.025 seconds/frame

4.2.2 Experiment 2: video Outputs

In the second experiment, the output from video is considered the video was created by stacked set of frames or images on top of each other that are chronologically arranged and hence show a slight change in localisation when moving from one frame to the next. Figure (4) shows video object can be detected and analysis of a minor accident, YOLO showed a much better performance in terms of bounding boxes for multiple cars and multiple people who are onlookers. MobileNet SSD on the other hand was only able to detect both cars in the same bounding box and failed to create a bounding box for the second onlooker. Hence, based on the precision, YOLO has a stronger performance against MobileNet SSD for videos with stationary objects.

In figure (5) illustrates the video object detection for moving objects, YOLO performs marginally better in terms of confidence.

The time taken to detect objects is of utmost importance here as well. The result demonstrates the latency time of MobileNet SSD model is smaller than YOLO.

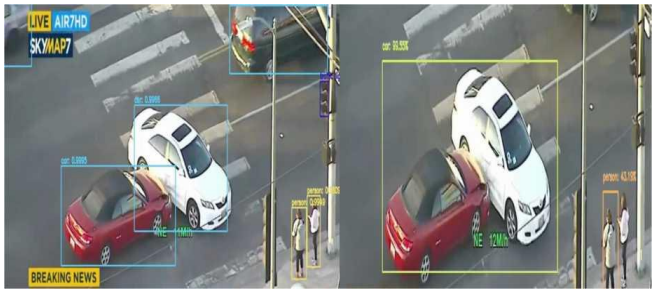

Figure (4): Video object detection for accident YOLO (left) and SSD (right)

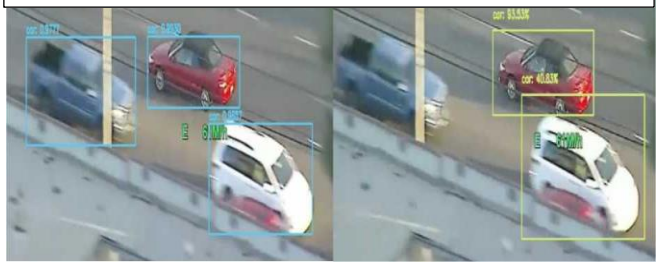

Figure (5): Video object detection for Moving objects YOLO (left) and SSD (right)

An amount of time was considered also for same set of above videos for each algorithm. The finding ravels that video with 23,722 frames was fed to the MobileNet SSD algorithm at it took about 10 minutes 22 seconds to analyze the entire video and provide the results while same video when fed to the YOLO algorithm took 2 hours and 58 minutes to analyses and provide the output.

In term of speed, YOLO is quite slow, and it was that the ~15x slower than MobileNet SSD due to, video being analyses by MobileNet SSD has taken in ten minutes whereas YOLO almost 3 hours to process.

Table 3: Time based comparison of YOLO, Faster R-MobileNet SSD for Experiment 1

Parameter	YOLO	MobileNet SSD
Frames in the video	23722	23722
Speed	Slow	Fast(Approximately 15 Times)
Time	0.4 Frame per Second	0.025 Frame Per Second

4.3 Results Comparison

The result of evaluation metrics for single image outputs and video outputs has been computed. We provided the result for both experimenters, we calculated the average value of each metric for first and second experiment respectively, the table (3) list the final result.

In terms of the precision and the F1 score, YOLO was the best model. It showed an average precision of close to 70 percent, while its F1 score was an even better 74.9%.

For the recall MobileNet SSD performed the best. It showed a recall of 86.68% and a MAP score of 82.31%. Although the YOLO achieves the best performance, the best appropriate algorithms should be able to detect images and videos containing small objects packed tightly together the YOLO object detector, is not well suited for small objects.

The finding also reveals that MobileNet SDD object detector, which offers an excellent balance of speed and precision. The MobileNet SDD is a combination of MobileNet with SSD. It can predict the kind and location of objects in a picture directly. It is easier to train in general and has been found to perform well on our database, even those including small objects.

Table 4: Results Comparison

Algorithm	Precision (%)	Recall (%)	F1 (%)	MAP (%)
MobileNet SSD	64.88	86.68	74.21	82.31
Faster R-CNN	60.11	73.89	66.29	78.23
YOLO	69.91	80.67	74.90	80.11

5. Conclusion and Future Work

The first experiment results shows that YOLO algorithm was exceedingly favourable, revealing the YOLO model's

extraordinary ability to detect moving object in images and draw anchor boxes around them with high confidence.

We used the MobileNet in conjunction with the SSD with the aim to reduce the computational cost and increase the speed

The result shows that MobileNet SSD was perform very well in team of computational cost

The whole process was much faster in producing the output than MobileNet SSD can detect many lesser objects SSD in real time. The result also demonstrated that model was able to discriminate between the left small objects a on the left and the motorcycle itself.

The Faster R-CNN can detect the objects exactly the same as YOLO, but the speed of object detection is just slightly better than that of YOLO and hence is a drawback. R-CNN is good at detecting small objects too, but the speed is an issue. It takes lot of time for execution and for rendering and training the image, whereas MobileMe SSD was extremely quick to do the same with decent accuracy.

The finding reveals the YOLO was able to process the video frame by frame very well and also, the confidence scores associated with it is also go, But the time taken to process the video was way too long and, in some cases, this could lead to a problem. The precision was about 70% and Recall was about 81% which is extremely good and is better than SSD and Fast R-CNN, but this is at the cost of high processing time. Additionally, detecting small objects also was not as reliable as MobileNet SSD.

Due to, the capacity of MobileNet SSD in reduce the number of parameters by utilizing the depth-wise separable convolution and spatial separable convolutions, The model capable to detect the small objects in videos very fast , Approximately 15 time faster than YOLO.

References

[1] A. A. and M.-M. P. E. Cervera-Uribe, "a deep learning approach for obstacle detection in self-driving cars," *Soft comput*, vol. 26, no. 11, pp. 5195–5207, 2022.

[2] C. , Z. Y. , G. Y. , S. M. R. and H. W. , Feng, "TOOD: Task-aligned One- stage Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 3490–3499. Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/2108.07755v3

[3] S. S. A. , A. M. S. , A. A. , K. N. , A. M. and L. B. , Zaidi, "A Survey of Modern Deep Learning based Object Detection Models. Digital Signal Processing: A Review Journal," 2022, Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/2104.11892v2

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020. [Online]. Available: http://arxiv.org/abs/2004.10934

[5] S. H. , A. C. and T. H. , aghavi, "Integrated real-time object detection for self-driving vehicles," IEEE, 2017, pp. 154–158.

[6] I. , K. S. , S. P. , W. M. A. , G. J. E. and S. I. Gog, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles.," in *IEEE International Conference on Robotics and Automation*, IEEE, 2021.

[7] S. , W. P. , F. A. K. , B. A. , F. C. and R. A. , Neumeier, "The holy grail to solve problems of automated driving? Sure, but latency matters," in *In Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* , 2019, pp. 186–197.

[8] M. A. Nees, "Acceptance of self-driving cars: An examination of idealized versus realistic portrayals with a self-driving car acceptance scale," in *Proceedings of the human factors and ergonomics society annual meeting*, Los Angeles: CA: SAGE Publications, 2016, pp. 1449–1453.

[9] C. , G. R. , C. R. V. , A. P. , C. V. B. , F. A. , J. L. , B. R. , P. T. M. , M. F. and de P. V. L. Badue, "Self-driving cars: A survey," *Expert systems with applications*, vol. 165, 2021.

[10] Venkata Satya Rahul Kosuru and Ashwin Kavasseri Venkitaraman, "Advancements and challenges in achieving fully autonomous self-driving vehicles," *World Journal of Advanced Research and Reviews*, vol. 18, no. 1, pp. 161–167, Apr. 2023, doi: 10.30574/wjarr.2023.18.1.0568.

[11] Dr. M. T., "Self Driving Car," *International Journal of Psychosocial Rehabilitation*, vol. 24, no. 5, pp. 380–388, Mar. 2020, doi: 10.37200/IJPR/V24I5/PR201704.

[12] Z. , Y. T. and X. L. , Yinghua, "Urban Street tree Recognition Method Based on Machine Vision," in *2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing, ICSP* , 2021, pp. 967–971.

[13] P. , Y. X. and G. Z. , Ping, "A Deep Learning Approach for Street Pothole Detection.," in *IEEE 6th International Conference on Big Data Computing Service and Applications, BigDataService 2020, pp.198-204.*, 2020.

[14] A. M. A. and D. G. G. M. ghani ABDULGHANİ, "Moving object detection in video with algorithms YOLO and faster R-CNN in different conditions," *Avrupa Bilim ve Teknoloji Dergisi, (33), pp.40-54*, vol. 33, pp. 40–54, 2022.

[15] F. Joiya, "OBJECT DETECTION: YOLO VS FASTER R-CNN," *International Research Journal of Modernization in Engineering Technology and Science*, Sep. 2022, doi: 10.56726/irjmets30226.

[16] N. Youssouf, "Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4," *Heliyon*, vol. 8, no. 12, Dec. 2022, doi: 10.1016/j.heliyon.2022.e11792.

[17] A. , T. N. H. , K. K. T. and H. C. S. . Ndikumana, "Deep learning based caching for self-driving cars in multi-access edge computing.," *IEEE Transactions*

on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2862–2877, 2020.

[18] C. , G. R. , C. R. V. , A. P. , C. V. B. , F. A. , J. L. , B. R. , P. T. M. , M. F. and de P. V. L. ,]Badue, "Fire hotspots detection system on CCTV videos using you only look once (YOLO) method and tiny YOLO model for high buildings evacuation.," in *international conference of computer and informatics engineering*, IEEE, 2019, pp. 87–92.

[19] P. , E. D. , L. F. , C. Y. and M. B. Jiang, "A Review of Yolo algorithm developments.," in *Procedia computer science*, 2022, pp. 1066-1073.

[20] O. Masurekar, O. Jadhav, P. Kulkarni, and S. Patil, "Real Time Object Detection Using YOLOv3," *International Research Journal of Engineering and Technology*, 2020, [Online]. Available: www.irjet.net

[21] S. M. and S. S. N. Abbas, "Region-based object detection and classification using faster R-CNN," in *4th International Conference on Computational Intelligence & Communication Technology (CICT)*, 2018, pp. 1–6.

[22] W. , Q. Y. and L. Y. , Chen, "An improved single shot detector for vehicle detection. Journal of Ambient Intelligence and Humanized Computing," *J Ambient Intell Humaniz Comput*, vol. 13, no. 11, pp. 5047-5053., 2022.

[23] B. Khasoggi, Ermatita, and Samsuryadi, "Efficient mobilenet architecture as image recognition on mobile and embedded devices," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 1, pp. 389–394, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.pp389-394.

[24] H. Nguyen, "A LIGHTWEIGHT AND EFFICIENT DEEP CONVOLUTIONAL NEURAL NETWORK BASED ON DEPTHWISE DILATED SEPARABLE CONVOLUTION," *J Theor Appl Inf Technol*, vol. 15, p. 15, 2020, [Online]. Available: www.jatit.org

[25] "A New Perspective for Mining COCO Dataset," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 80–89, Sep. 2023, doi: 10.33103/uot.ijccce.23.3.7.

[26] T. Y. , M. M. , B. S. , H. J. , P. P. , R. D. , D. P. and Z. C. L. Lin, "Microsoft coco: Common objects in context.," in *In Computer Vision–ECCV 2014: 13th European Conference*, , Springer International Publishing., Ed., Zurich, 2014, pp. 740–755.