Received: September 5, 2023 | Revised: December 12, 2023 | Accepted: December 13, 2023

JOURNAL OF APPLIED RESEARCH ON SCIENCE AND TECHNOLOGY
JARST
INSTITUTE OF RESEARCH AND DEVELOPMENT
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI
ISSN ONLINE: 2773-9473
https://ph01.tci-thaijo.org/index.php/rmutt-journal

Real-time vehicle detection system on the highway

Pisanu Kumeechai*

Engineering Department, Education Branch, Royal Thai Naval Academy, Samut Prakan 10270, THAILAND

*Corresponding author: Pisanu41984198@hotmail.com

ABSTRACT

Locating and classifying different types of vehicles is a vital element in numerous applications of automation and intelligent systems ranging from traffic surveillance to vehicle identification, with deep learning models now dominating the field of vehicle detection. However, vehicle detection in Bangladesh remains a relatively unexplored research lacuna. One of the main goals of vehicle detection is its real-time application, with "You Only Look Once" (YOLO) models proving to be the most effective. This paper compared real-time vehicle highway detection systems using YOLOv4, Faster R-CNN and SSD algorithms to determine the best performance. A vehicle detection and tracking system was also developed that improved highway safety. Vehicle trials compared the real-time performances of the YOLO, Faster R-CNN and SSD algorithms in detecting and tracking highway vehicles by measuring precision, recall, F1-score and operating speed. Models for each algorithm were constructed and each model was trained and tested, with performance measured using a confusion matrix. This statistical tool assessed the efficiency of the system using a prepared test dataset and evaluated the results using appropriate indicators such as real-time road lines, traffic signs and vehicle detection false positive rates. Results showed that the YOLOv4 algorithm outperformed Faster R-CNN and SSD in real-time vehicle detection and tracking on highways. YOLOv4 also processed the results more quickly and proved superior in detecting and tracking objects in real time. The Faster R-CNN algorithm gave high object detection, tracking accuracy and recall while reducing the number of locations needing detection, with the SSD algorithm providing high precision, recall and good image detection results.

Keywords: Real-time vehicle detection, YOLOv4, Faster R-CNN, SSD, Highway safety

INTRODUCTION

Real-time detection and tracking of vehicles on highways have benefited from high-profile developments in image processing over the past decade. Vehicle detection and tracking systems are critical for maintaining highway safety by quickly identifying obstructions and traffic violations to control vehicle speed limits in the event of accidents. Cutting-edge technologies such as image processing and deep learning are now utilized to develop vehicle detection and tracking systems, delivering safety and efficiency in highway traffic. This research proposed a real-time vehicle detection and tracking system using image processing and deep learning algorithms. The developed system detected vehicles moving on highways in real time and accurately tracked them using the three experimental algorithms YOLOv4, Faster R-CNN and SSD. Each algorithm has advantages and disadvantages. YOLO (You Only Look Once) [1-3] is a high-speed algorithm that works in split-image mode (single-stage), making it possible to detect objects quickly and YOLO can detect larger objects better than other algorithms. However, YOLO has difficulties detecting small objects with oblique details, as shown in figure 1. The hallmark of Faster R-CNN [4-6] is accuracy and efficiency in simultaneously detecting multiple objects with oblique details. This algorithm is suitable for tasks that require detecting and identifying multiple objects in a single image. One disadvantage of Faster R-CNN is its slow performance, generated by a complex workflow. The area containing the object must first be extracted before predicting the memory required to detect the object. Faster R-CNN is slower than algorithms that do not have to perform this step, such as YOLO. The SSD (Single Shot MultiBox Detector) algorithm [7] can detect objects moving at high speed using a split image function that works in real time. This algorithm uses a data pyramid to detect multiple small objects in the image simultaneously. However, the SSD algorithm has difficulty detecting large objects with oblique details, object movements, or changes in perspective.

Several recent studies have addressed the problem of vehicle detection using aerial imagery [8-11]. Previously [12], we compared YOLOv3 with Faster R-CNN for vehicle detection from aerial imagery using a small dataset of low-altitude UAV images collected

DOI: 10.60101/jarst.2023.254011

on the premises of Prince Sultan University. Imaging altitude plays an important role in accuracy detection. Advanced performance indicators include Intersection over Union (IoU) and mean Average Precision (mAP). This article considers several datasets with different configurations and presents a comprehensive comparison analysis between three state-of-the-art approaches: Faster R-CNN, YOLOv3 and YOLOv4. The challenges faced by photo physics were discussed in ref [13]. Problems in airborne vehicle detection include small objects and complex backgrounds. These problems were solved using neural networks. Other research problems when applying deep learning techniques to aerial imagery have been discussed in various contexts, including object detection and classification [14,15], semantic segmentation [16-18] and generative adversarial networks (GAN) [19]. Jiao et al. [20] surveyed many object detectors and reported results on a COCO dataset [21]. This study focused on an in-depth comparison of three recently published algorithms representing two main types of object detectors, Faster R-CNN [22] (a two-stage detector) and YOLOv3 [23] and YOLOv4 [24] (single-step detectors), to investigate a wide range of hyperparameters and evaluate the impact of size and characteristics of aerial view datasets.

This paper proposes a method that obtains the driving area of a vehicle through road line detection and captures the main color information of the motion area and non-motion area of a moving vehicle. The experimental algorithms YOLOv4, Faster R-CNN and SSD were used to detect and track the vehicles. The vehicle detection and tracking model is shown in figure 2. The detector is responsible for detecting the vehicles in each frame, while the tracker correlates vehicles in adjacent frames to form a complete vehicle trajectory.

The rest of this paper is organized as follows: Section II briefly reviews the existing object detection algorithms used for vehicle detection and tracking. Section III explains the method that applied in this paper. Section IV explains the use of a confusion matrix for measurement. The experiments and comparative analyses to evaluate the state-of-the-art methods are detailed in Section V and the conclusions drawn are presented in Section VI.

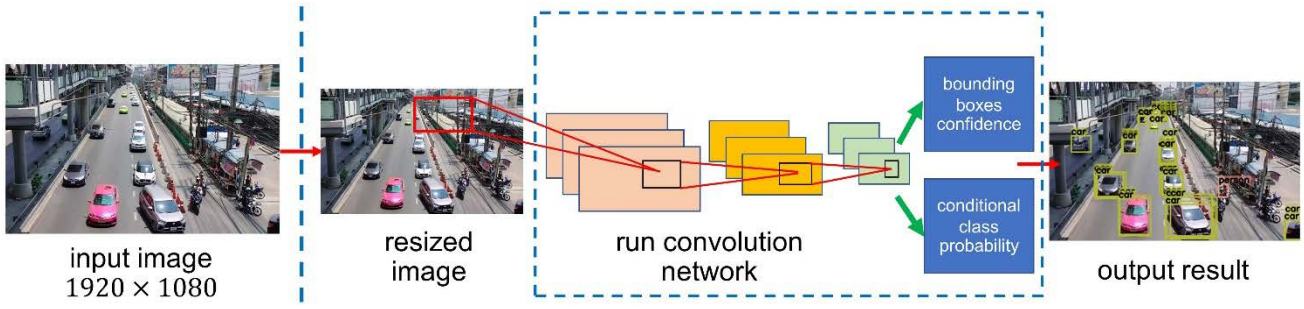


Figure 1 Working principle of YOLO.

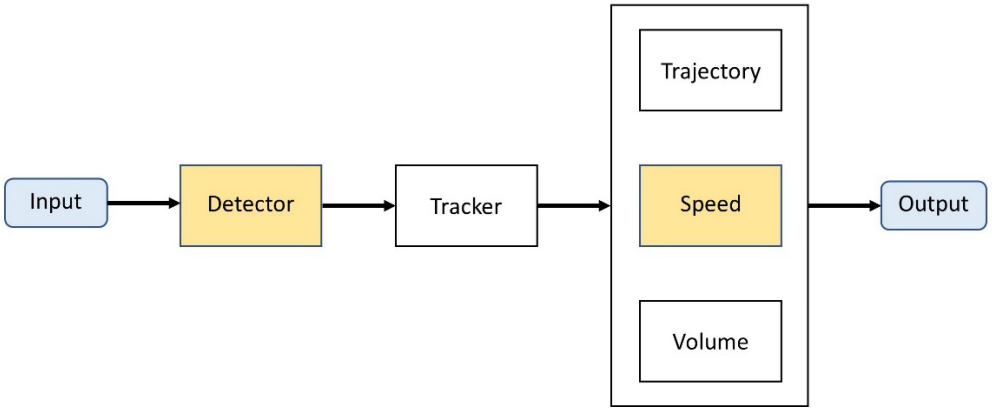


Figure 2 The detection-based vehicle tracking model.

MATERIALS AND METHODS

Research scope

The research objective was to compare the YOLOv4, Faster R-CNNF and SSD algorithms to determine the best performance and develop a vehicle detection and tracking system to improve highway safety. Digital cameras with image file sizes 640 by 840 and 50 frames per second were used to collect the information. The image dataset detected images from normal, well-lit surroundings and divided 1000 images into 800 training images and 200 testing images, representing different vehicles such as cars, trucks, motorcycles and buses to evaluate and measure the effectiveness of the system in detecting and tracking vehicles in actual traffic conditions.

The trial compared the performance of YOLO, Faster R-CNN, and SSD in detecting and tracking highway

DOI: 10.60101/jarst.2023.254011

vehicles by measuring precision, recall, F1-score, and operating speed. The accuracy and performance of each algorithm were assessed in the same situation.

This real-time vehicle detection and tracking system on highways used YOLO, Faster R-CNN and SSD algorithms and tuning parameters such as detector resolution to achieve a high-performance system in a highway environment for real-world use.

Related literature

1. Vehicle detection

Real-time detection of vehicles on highways represents an important field of study with benefits in many areas, including the prevention of road accidents. Highway vehicle detection and tracking is an integral part of the road safety system, covering events on the highway, such as changing lanes. Traffic control or vehicle problem detection to reduce road accidents using traffic management. Highway vehicle detection and tracking systems effectively manage vehicle speed while also controlling highway lighting. Data collection from highway vehicle detection and tracking was analyzed to gain insights into traffic trends, risks and factors affecting highway safety as a research area with potential for further development. Copious previous research has addressed these issues. Njayou Y, [25] used Convolutional Neural Network (CNN) image processing techniques for traffic sign classification and object detection with Faster R-CNN and YOLOv4 algorithms. The Faster R-CNN test results gave a mean average precision (mAP) of 43.26% at 6 frames per second (fps), which was unsuitable for real time use. By contrast, YOLOv4 yielded mAP of 59.88% at 35 fps as a suitable model for real-time traffic sign detection. Annam F, et al. [26] proposed a fast and accurate real-time vehicle detection method using deep learning techniques for unrestricted environments. They detected vehicles in the model image and real environment using deep neural networks as the main tool. The dataset consisted of vehicle images taken in various environments and was divided into training and test sets. The proposed detection method gave high speed and accuracy. The concept of the Faster R-CNN model used the feature extraction technique and depth processing model. Results demonstrated the efficiency of the proposed method in terms of vehicle detection accuracy and short image processing time.

Bochkovskiy A, et al. [27] proposed the YOLOv4 algorithm as a Single Shot Detector (SSD) object detection model that focused on providing optimal speed and accuracy for object detection using a combination of techniques to improve YOLOv3 and the existing models. They added depth and used new techniques to increase the efficiency and accuracy of the system. Results demonstrated that YOLOv4 detected objects in images with high speed and accuracy. Ren S, et al. [28] used real-time Region Proposal Networks (RPN) in the Proposal Generation process by combining CNN and RPN into a single structure, enabling fast and highly efficient rapid object detection. This new concept created images based on deep neural networks and then used these images for object detection. Results demonstrated the high speed and accuracy of Faster R-CNN in object detection. Liu W, et al. [29] proposed an SSD (Single Shot MultiBox Detector) algorithm using rapid duplex object detection. This method combined the detection and ranking of several class objects into a single structure, enabling fast and highly efficient detection of objects that did not require new proof procedures.

2. Vehicle tracking

CNN-based multi-target vehicle tracking technology has recently received increased attention [30]. Bewley et al. [31] used the Faster R-CNN algorithm as the target detector and proposed a simple online and real-time tracking (SORT) algorithm to track multiple targets simultaneously based on the Kalman filter and Hungarian matching algorithm, while Wojke N, et al. [32] considered both the movement characteristics and the appearance of the target. They proposed an improved DeepSORT algorithm where the appearance features of the target were extracted through the CNN model after target detection by the detector. Later, Wang Z, et al. [33] further improved the complexity of the DeepSORT algorithm by developing the JDE algorithm. This directly exported the target location and appearance features into the detection network using RetinaNet to embed an instance-level vehicle feature extraction network in the detector model, while Lu Z, et al. [34] created a RetinaTrack multi-target vehicle tracking model to combine vehicle movement and appearance characteristics for data association.

Method

YOLOv4 is a high-speed, high-precision rapid-duplex object detection model that uses deep neural networks (Convolutional Neural Network) to detect objects in images YOLOv4 includes Backbone: CSPDarknet53, Neck: SPP, PAN, Head: YOLOv3 and uses Bag of Freebies (BoF) and Bag of Specials (BoS) to optimize the model. BoF includes CutMix and Mosaic data augmentation, DropBlock regularization and Class label smoothing, while BoS includes Mish activation, Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC), SPP-block, SAM-block, PAN path-aggregation block and DIoU-NMS to increase model precision and speed.

YOLOv4's object detection and tracking algorithm uses a CNN (Convolutional Neural Network) structure that comprises the backbone. CSPDarknet53 is used as the core structure of the YOLOv4 model to extract image features. Neck uses SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network) to increase object detection accuracy, and the head uses YOLOv3 (anchor-based) as the head of the YOLOv4 model to perform object detection.

DOI: 10.60101/jarst.2023.254011

To perform object detection and object tracking functions, YOLOv4 uses an improved CNN structure over YOLOv3 with the addition of Bag of Freebies (BoF), a model modification technique that is not a core part of the YOLOv4 model but enhances model accuracy. BoF includes CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing, and Bag of Specials (BoS) to increase accuracy.

In YOLOv4, the input image is split into a grid S×S, with each grid cell in charge of vehicle detection. B bounding boxes are placed in each grid cell and the network then outputs an offset value for the bounding box and class probability. The bounding boxes are chosen and the vehicle in the image is then located using those with a class probability with a particular threshold (Figure 3).

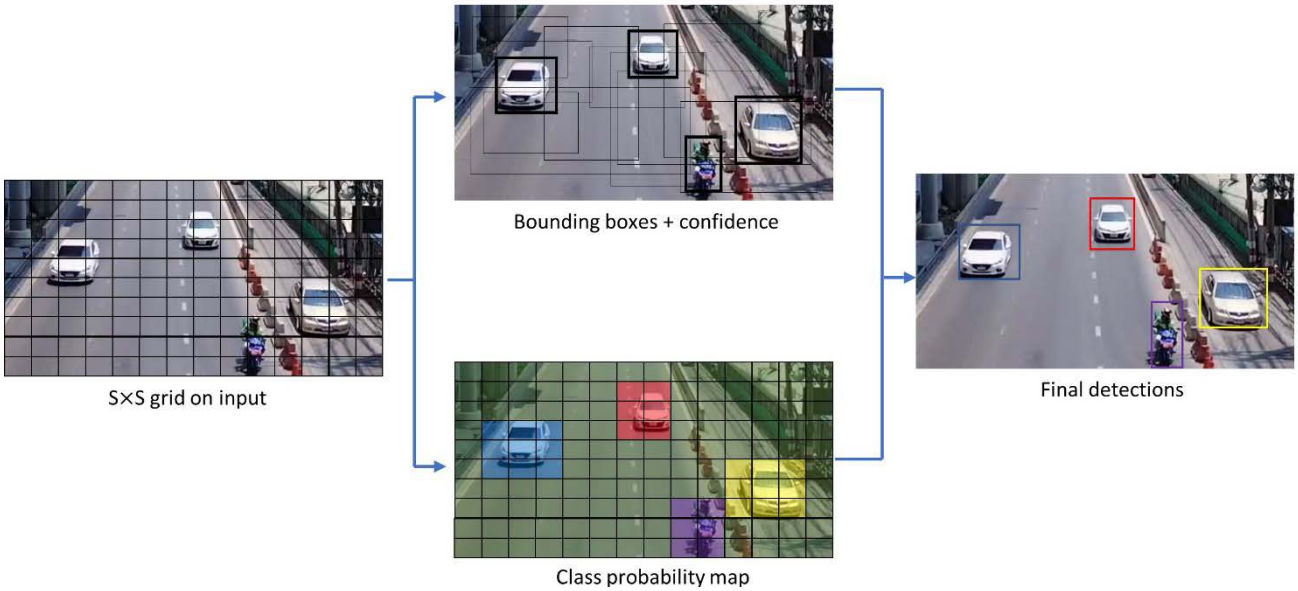


Figure 3 Illustration of YOLOv4 framework.

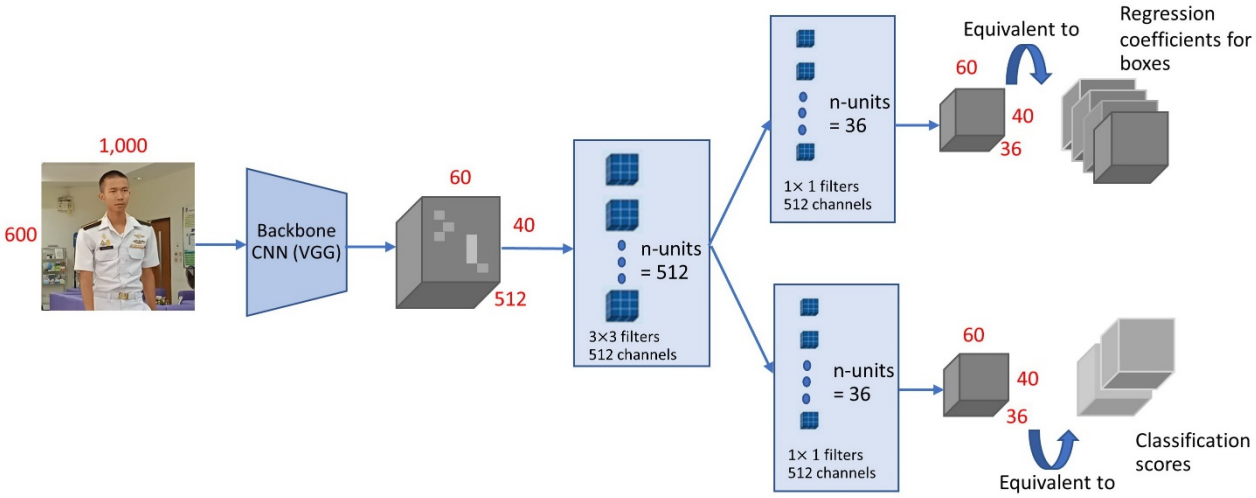


Figure 4 Faster R-CNN architecture.

YOLOv4 is a fully convolutional network with 110 convolution layers comprising 66 as 1 × 1 and 44 as 3 × 3. The input layer has a 3 × 3 convolution layer with 19 32 filters. This thesis used input size starting at 416 × 416 with 3 channels (RGB). The output layer is a 1 × 1 convolution layer with a stride and padding size of 1. The output layer has 33 filters and uses CSPDarknet53 as the backbone, SPP and PAN for the Neck, with the head of YOLOv4 used as the head in YOLOv4. Mini-batch gradient descent with momentum is used for optimization. CSPDarknet-53 extracts deep features of the input images, while SPP efficiently increases the receptive field, PANet extracts features across multiple scales and the heads detect objects. A linear activation function is used for the final layer, with input starting size 416 × 416 with 3 channels (RGB), YOLOv4 has more than 60M parameters.

Faster R-CNN [36] (Region-based Convolutional Neural Network) is an algorithm designed to detect objects and track images quickly with high accuracy (Figure 4). The algorithm can be divided into three parts as follows:

1) Faster R-CNN uses a Region Proposal Network (RPN) network to create a region proposal

DOI: 10.60101/jarst.2023.254011

as a region that may have straightforward objects. The pre-trained feature map is used as an assembly.

2) Region of Interest (RoI) pooling. Upon receiving an image proposal from an RPN, Faster R-CNN uses RoI pooling to adjust the size and position of the image proposal to fit the deep neural network for tracking the object.

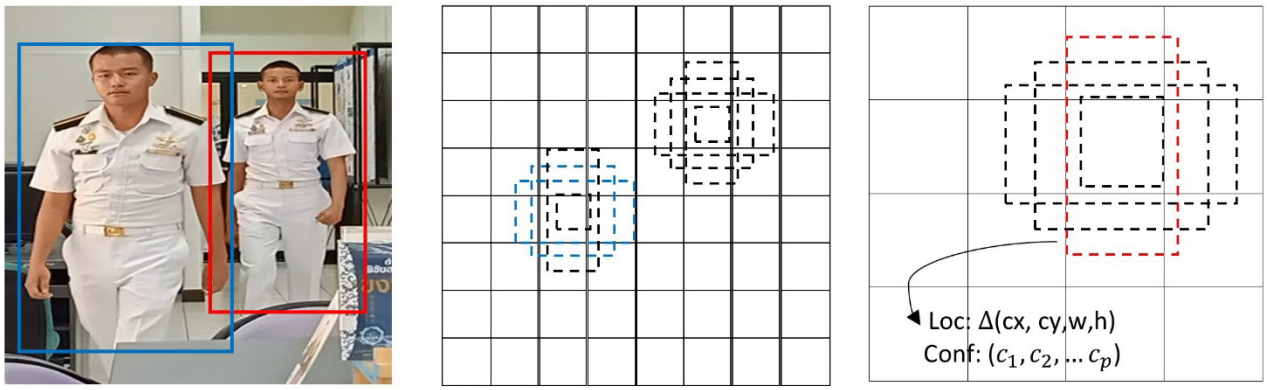
3) Region Classification and Regression. The last step of the Faster R-CNN deep neural network classifies objects and adjusts the size and position of the bounding boxes found in the proposed segments of the image. Object classification is the process used to decide the type of object while adjusting the size and position of the squares to improve object detection accuracy.

Faster R-CNN is the most preferred and used version of the R-CNN family, using a particular selection of search algorithms for proposing regions and taking only 1 or 2 seconds per image to run the CPU computation. Faster R-CNN uses Region Proposal Networks (RPNs) to generate region proposals, reducing generation time from seconds to milliseconds per image.

In FASTER R-CNN,

- RPN generates bounding boxes as a rectangular box that surrounds an object and specifies its position, class (e.g. car, person) and confidence (how likely it is to be at that location).
- In this stage, CNN is used to generate features of these objects. Region proposal is not performed on the original image but on the final feature image, which is then input into the ROI pooling (Region of Interest Pooling fixes image size requirements for object detection).
- The output from the ROI pooling layer has a size of N, 7, 7, 512, where N is the number of proposals from the region proposal algorithm. After passing those ROI pooling outputs through two fully connected layers, the features are fed into the sibling classification and regression branches.
- A classification layer is present to determine the object's class.
- Finally, a regression layer is used to ensure that the coordinates of the bounding boxes are more precise, leaving no gaps for errors.
- Anchors are introduced in RPN to deal with different scales and aspect ratios of the objects. An anchor is positioned at each sliding location of the convolutional maps and at the center of each spatial window. Each anchor is associated with a scale and an aspect ratio.

SSD (Single Shot MultiBox Detector) [37] is a method of object detection and image tracking using a single deep neural network, which is shown in figure 5. The SSD divides the regions of the bounding boxes into a set of default boxes. The aspect ratio and scales differ for each feature map location. To hold objects of different sizes and shapes, the network generates a score for factoring each object category in each initial box and generates adjustments to make the box fit the shape of the object. Predictions from multiple property maps with different resolutions are used to handle objects of different sizes. SSDs have been tested, with results proving that increasing the number of carefully selected default boxes greatly improves object detection and tracking performance. SSD is a highly efficient and easy-to-use object detection and tracking model.



(a) Image with GT boxes (b) 8 × 8 feature map (C) 4 × 4 feature map

Figure 5 SSD framework.

(a) SSD only needs an input image and ground truth boxes for each object during training. A small set (e.g. 4) default boxes of different aspect ratios is evaluated in a convolutional fashion at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in figure 5 (b) and (c)).

The key difference between training an SSD and a typical detector that uses region proposals is that ground truth information must be assigned to specific outputs in the fixed set of detector outputs. Some versions also require training in YOLO [38] and the region proposal stage of Faster R-CNN [39] and MultiBox [40]. Once this assignment is determined, the loss function and backpropagation are applied end-to-end. Training also involves choosing the set of default

DOI: 10.60101/jarst.2023.254011

JOURNAL OF APPLIED RESEARCH ON SCIENCE AND TECHNOLOGY

boxes and scales for detection as well as hard negative mining and data augmentation strategies.

Training Objective. The SSD training objective is derived from the MultiBox objective [23] but is extended to handle multiple object categories. Let $x_{ij}^{p} = \{1,0\}$ be an indicator to match the i-th default box to the j-th ground truth box of category p. In the matching strategy above, $\sum_i x_{ij}^{p} \geq 1$. The overall objective loss function is a weighted sum of the localisation loss (loc) and the confidence loss (conf) such that:

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right) \quad (1)$$

where N is the number of matched default boxes and the localization loss is the Smooth L1 loss [41] between the predicted box (l) and the ground truth box (g) parameters. Like Faster R-CNN [42], we regress to offsets for the center of the bounding box and its width and height. Our confidence loss is the SoftMax loss over multiple class confidences (c) and the weight term α is set to 1 by cross-validation.

4. Vehicle tracking method

Section III describes the Simple Online and Real-time Tracking (SORT) algorithm used to track an individual vehicle. SORT is a real-time and online tracking algorithm integrated with the Kalman filter and the Hungarian algorithm, with an accuracy comparable to state-of-the-art online trackers while supporting higher update rates. The state space of each vehicle is modeled as follows: $[x,y,s,r,\dot{x},\dot{y},\dot{s}]^{T}$, where (x, y) is the central location of the bounding box of the vehicle, s and r represent the scale (area) and the aspect ratio of the bounding box respectively and $\dot{x}$ and $\dot{y}$ are the velocity elements. When detection is associated with a vehicle, the detected bounding box is used to update the state of the vehicle. The location of the new bounding box for each vehicle is predicted based on the Kalman filter. SORT uses the intersection-over-union (IOU) between each incoming detection and all predicted bounding boxes of the existing vehicle. These values are then used to populate a cost matrix solved by a Hungarian algorithm to assign each track an appropriate detection. The vehicle tracking procedure of SORT is shown in figure 6.

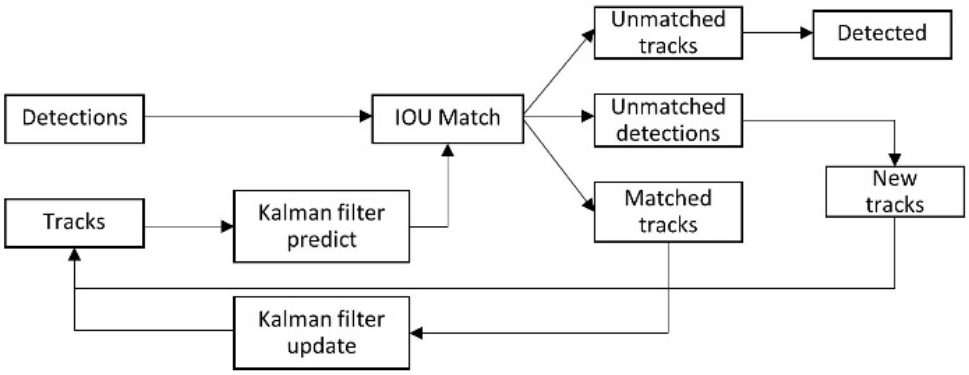


Figure 6 Vehicle tracking procedure of SORT.

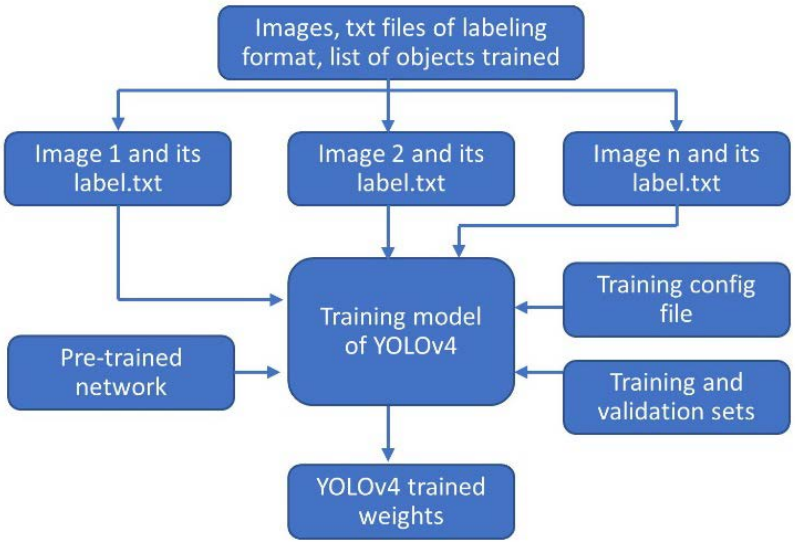


Figure 7 Training diagram of YOLOv4 on our dataset.

Confusion matrix theory

Confusion matrix theory is a statistical tool used to evaluate the effectiveness of classification systems, especially binary systems, by considering the predicted result against the ground truth of the sample data (rows) and the actual values (columns). Each cell in the confusion matrix represents the number of examples

DOI: 10.60101/jarst.2023.254011

correctly or incorrectly classified for each group of a sample type. The classification results can be divided into four parts, according to Table 1.

Table 1 Confusion matrix.

	Predicted positive	**Predicted negative**
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

The variables used in the confusion matrix are defined below.

1) True Positive (TP) is the number of samples predicted to be positive and truly positive.

2) False Positive (FP) is the number of samples predicted to be negative but are actually positive.

3) False Negative (FN) is the number of samples predicted to be positive but are actually negative.

4) True Negative (TN) is the number of samples predicted to be negative and true negative.

The TP, FP, FN and TN values were used to calculate system efficiency, with the important equations 2, 3 and 4 shown below.

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q} AP_q \quad (5)$$

Precision is the accuracy of forecasting.

Recall is the accuracy of positive detection.

The F1 score is the mean of precision and recall and is used to evaluate the performance of a particular binary classification system.

The mAP is the average value of AP for each class.

Research process

This research used the following steps to compare the performance of YOLOv4 (Figure 7). Faster R-CNN and SSD real-time highway vehicle detection and tracking algorithms.

1) Set the objectives

The research objective was to develop a highway detection and tracking system to enhance the efficiency and accuracy of vehicle detection and increase road safety.

2) Create a dataset

Building the suitable dataset for training and testing the system involved taking photographs or videos of the highway vehicles being tracked to increase detection accuracy. The dataset was divided into two parts the training dataset and the testing dataset.

3) Train and test the model

A built-in dataset was employed to train and test the models used in vehicle detection and tracking using the YOLOv4, Faster R-CNN and SSD algorithm training methods on the defined dataset.

4) Measure the performance

The performance was measured using a confusion matrix after successfully training and testing the models for each algorithm. This statistical tool assessed the efficiency of the system using the prepared test dataset and evaluated the results using appropriate indicators such as detection rate and false positive rate.

RESULTS AND DISCUSSION

The dataset and evaluation index

Deep learning-based vehicle object detection algorithms learn features from data samples and the dataset must be representative. The dataset contained real image data collected from scenes such as urban areas, rural areas and highways. Each image contained up to 15 cars and 30 pedestrians, with various degrees of occlusion and truncation. The images reflected a variety of complex situations, such as multiple trucks and cars appearing in the same image or multiple trucks, vans and cars appearing in the same image under different lighting conditions, road environments and road conditions. For the peculiar application of vehicle object detection, the Car, Van, Truck, Pedestrian, Pedestrian (sitting), Cyclist, Tram and Misc classes in the dataset were converted to Car, Van, Truck and Other classes. The 1000 labeled images in the dataset were divided into a training set and a testing set at ratio 6:4. To validate the generalization ability of the algorithm, a vehicle dataset was established through online collecting and real-scene shooting. Figure 8 shows some samples from the dataset.

Experimental setup

The experiment was run in a Python environment using Google Collaboratory. The average usable memory of the machine was 8 GB and the average disk space used was 256 GB. The sample images were randomly split into 60% for training, 20%, for validation and 20% for testing. The batch size was 16 for all the models because smaller batches are noisy and can reduce generalization errors. Model Checkpoints were used to save the model with the best validation accuracy during the process. The convergence of the models depended on many parameters and the size of the weight parameters. Using the Early Stopping technique, the total time required to complete the training of all three models was more than 500 hours.

Research results

Small batch random gradient descent was used for training optimization and a set of values that improved

DOI: 10.60101/jarst.2023.254011

JOURNAL OF APPLIED RESEARCH ON SCIENCE AND TECHNOLOGY

the network quality was selected. The momentum parameter was set to 0.8, the weight attenuation coefficient was set to 0.0001 and the initial learning rate was set to 0.001. Due to memory limitations, the batch was set to 64 and the subdivision was set to 64. Two indicators were considered to verify the performance of each vehicle detection algorithm detection accuracy and detection efficiency. Comparative experiments were conducted to prove that each part of the modified YOLO was effective.

Real-time detection and tracking of vehicles on highways were conducted using YOLOv4, Faster R-CNN and SSD algorithms to determine the best performance, with research findings summarized in Tables 2 and 3.

Table 2 Confusion Matrix Measurement Results.

	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)
YOLOv4	90	10	5	895
Faster R-CNN	85	15	7	893
SSD	88	12	8	892

Table 3 Precision, Recall and F1-score.

	Average IoU	Precision	Recall	F1-score
YOLOv4	93.17%	0.900	0.947	0.923
Faster R-CNN	85.86%	0.850	0.923	0.885
SSD	80.11	0.880	0.916	0.898

Detecting image objects was assigned into four categories. Each category had a True Positive (TP) value as the number of objects that were correctly detected and tracked by category, False Positive (FP) value as the number of objects that were detected but not tracked or incorrectly tracked by category, False Negative (FN) value as the number of objects that were present but were not detected or incorrectly detected as a category and True Negative (TN) value as the number of objects that were not objects in a category and were not detected as objects in a category (Table 2). YOLOv4 gave a good performance with the highest value of True Positive (TP) and the lowest value of False Negative (FN). The TP, FP, FN and TN values were then used to calculate the Precision, Recall and F1-score, including other values related to the image detection and tracking system performance, as shown in Table 3.

Real-time detection and tracking results of highway vehicles are shown in Table 3. In table 3, the YOLOv4 algorithm had Precision of 0.900, Recall of 0.947 and F1-score of 0.923, indicating good performance in detecting all images. The Faster R-CNN algorithm had Precision of 0.850, Recall of 0.923 and F1-score of 0.885, indicating good performance in image detection. The Faster R-CNN algorithm had the highest Recall but the Precision and F1-scores were lower than YOLOv4. The SSD algorithm had Precision of 0.880, Recall of 0.916 and F1-score of 0.898, indicating good performance in image detection. The SSD algorithm had relatively high Precision and Recall and a high F1-score. When comparing the Precision, Recall and F1-scores of all three algorithms, the YOLOv4 algorithm gave the best performance in image detection.

The faster R-CNN algorithm and SSD algorithm showed similar performances but SSD had higher precision and recall. The real-time vehicle detection and tracking system on the highway is shown in figure 8.

Figure 8 Real-time vehicle detection and tracking system.

In Table 3, when the IoU threshold of the YOLOv4 model trained on the Royal Thai Naval Academy 1000 dataset was changed from 0.5 to 0.75, the average IoU was 47.08%. The F1-Score of YOLOv4 was 0.923 when the IoU threshold was 0.5 but was relatively biased, with Faster R-CNN and SSD as 0.885 and 0.898, respectively, when the IoU threshold was 0.75. Thus, some classes were better learned than others. As shown in Table 3, after the feature extraction network of the original YOLOv4 was replaced by the skip-connection deep residual feature extraction network, in Table 3, the recall and mAP increased from Faster R-CNN of 7.31% and from SSD of 13.06%, respectively, because the feature extraction network proposed in this paper reused the features extracted from the input feature.

Additional experiments were conducted with different values of learning rates (10^{-5}, 10^{-4}, 10^{-3} and

DOI: 10.60101/jarst.2023.254011

10^{-2}) for each of the main algorithms (Faster R-CNN with Inception-v2, Faster R-CNN with Resnet 50 and YOLOv4 with input size 608 × 608) to measure the sensitivity of each algorithm to the learning rate hyperparameter. A learning rate of 10^{-3} yielded the best performance in most cases but on the Stanford dataset Faster R-CNN with Inception-v2 and YOLOv4 gave better results at lower learning rates (10^{-4} and 10^{-5}, respectively). A learning rate of 10^{-2} gave poor results in all cases except for YOLOv4 on both datasets and for Resnet50 on the dataset. A learning rate of 10^{-1} was also tested but this led to a divergent loss. These results highlighted the importance of testing different learning rate values when comparing object detection algorithm performances. Results in Table 3 and Table 4 confirm the superior performance of YOLOv4 and Faster R-CNN when the learning rate was well chosen.

Table 4 Detailed results of different configurations of YOLOv4, SSD and Faster R-CNN, on the Stanford dataset.

Algorithm	Feature Extraction	Input Size
YOLOv4	CSPDarknet-53	320 × 320 (fixed)
YOLOv4	CSPDarknet-53	416 × 416 (fixed)
YOLOv4	CSPDarknet-53	608 × 608 (fixed)
Faster R-CNN	Inception v2	600 × 816 (variable)
Faster R-CNN	Inception v2	608 × 608 (fixed)
Faster R-CNN	Resnet50	600 × 816 (variable)
SSD	Darknet-53	320 × 320 (fixed)
SSD	Darknet-53	416 × 416 (fixed)
SSD	Darknet-53	608 × 608 (fixed)

For YOLOv4, one-stage detectors performed well over a dense selection of potential object locations, enabling whole image detection in a single forward propagation. Two-stage detectors such as Faster R-CNN are generally more accurate than their one-stage counterparts but fast speeds and low memory costs render one-stage detectors more suitable for real-time applications such as autonomous driving.

CONCLUSION

YOLOv4 outperformed Faster R-CNN and SSD in real-time vehicle detection and tracking on highways because YOLOv4 processed the results faster and was ideal for detecting and tracking objects in real time, as a key feature in detecting high-speed highway vehicles. YOLOv4 offered high object detection and tracking accuracy by precisely identifying the location and size of the vehicle. This is important for managing and improving highway traffic. However, in some cases, YOLOv4 encountered difficulty or finely detailed objects, causing some detection discrepancies. Therefore, YOLOv4 may experience reduced performance in challenging environments.

Faster R-CNN had high object detection, tracking accuracy and recall. The system detected and tracked a wide range of objects in the test dataset by creating region proposals that accurately determined object boundaries. This reduced the number of locations that needed to be detected. However, the Faster R-CNN algorithm was executing slower because it had to create region proposals before performing object detection. This resulted in uptime that could limit fast deployment in environments that require immediate response.

The SSD algorithm gave high precision and recall and good image detection performance, with a high ability to detect small objects such as small rings or turn signals. However, limitations of SSD included accuracy in detecting large objects with fine detail.

Recommendations to improve real-time detection and tracking of vehicles on highways using YOLOv4 include: 1. Increase the amount of training data used to train YOLOv4 to improve the accuracy and detection of the system 2. Experimentally tune YOLOv4 parameters such as detection thresholds or filters to suit the dataset and objects to be detected 3. Increase YOLOv4 resolution by adding or enhancing model layers to detect and track objects in more detail.

Suggestions for further research

Real-time detection and tracking of vehicles on highways using the YOLOv4 algorithm can be improved as follows:

1) Customize and add training data: Increase the amount of training data used to train YOLOv4 to improve the accuracy and detection of the system. Consider adding information such as different weather conditions and lighting to improve system operation.

2) Parameter customization hyperparameter tuning. Experimentally tune YOLOv4 parameters such as detection thresholds or filters to suit the dataset and objects to be detected. Parameter tuning may achieve better results in vehicle detection and tracking.

3) Model refinement: Consider increasing the resolution in YOLOv4 models by adding or enhancing model layers to detect and track objects in more detail. For example, overlapping vehicle sections can be detected and tracked.

ACKNOWLEDGEMENT

This research was successfully accomplished with the assistance of the Engineering Department, Education Branch, Royal Thai Naval Academy, which provided both equipment and research facilities.

REFERENCES

1. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer

DOI: 10.60101/jarst.2023.254011

vision and pattern recognition. 2016;2016:779-788. doi: 10.1109/CVPR.2016.91.

2. Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv preprint arXiv. 2018; doi: 10.48550/arXiv.1804.02767.
3. Bochkovskiy A, Wang CY, Mark Liao HY. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv. 2020;2004.10934.
4. Ren S, He K, Girshick R, Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems. 2015;91-9.
5. Girshick R. R-CNN F. In Proceedings of the IEEE international conference on computer vision. 2015; 2015:1440-8.
6. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39(6): 1137-49.
7. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Berg. SSD: Single shot multibox detector. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018;40(4):817-23.
8. Li X, Luo M, Ji S, Zhang L, Lu M. Evaluating generative adversarial networks-based image-level domain transfer for multi-source remote sensing image segmentation and object detection. Int J Remote Sens. 2020;41:7327-51. doi:10.1080/01431161.2020.1757782.
9. Liu K, Mattyus G. Fast multiclass vehicle detection on aerial images. IEEE Geosci Remote Sens Lett. 2015;12:1938-42. doi: 10.1109/LGRS.2015.2439517.
10. Audebert N, Le Saux B, Lefèvre S. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. Remote Sens. 2017;9:368. doi:10.3390/rs9040368.
11. Ma B, Liu Z, Jiang F, Yan Y, Yuan J, Bu S. Vehicle detection in aerial images using rotation-invariant cascaded forest. IEEE Access. 2019;7:59613-23.
12. Benjdira B, Khursheed T, Koubaa A, Ammar A, Ouni K. Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. In Proceedings of the 2019 IEEE 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5-7 February 2019; p.1-6.
13. Xi X, Yu Z, Zhan Z, Tian C, Yin Y. Multi-task cost-sensitive-convolutional neural network for car detection. IEEE Access. 2019;7:98061-8. doi: 10.1109/ACCESS.2019.2927866.
14. Ševo I, Avramovi'c A. Convolutional neural network based automatic object detection on aerial images. IEEE Geosci Remote Sens Lett. 2016;13: 740-4. doi: 10.1109/LGRS.2016.2542358.
15. Ochoa KS, Guo Z. A framework for the management of agricultural resources with automated aerial imagery detection. Comput Electron Agric. 2019; 162:53-69. doi: 10.1016/j.compag.2019.03.028.
16. Kampffmeyer M, Salberg A, Jenssen R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June-1 July 2016; p. 680-688. doi: 10.1109/CVPRW.2016.90.
17. Azimi SM, Fischer P, Körner M, Reinartz P. Aerial LaneNet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks. IEEE Trans Geosci Remote Sens. 2019;57:2920-38. doi: 10.1109/TGRS.2018.2878510.
18. Mou L, Zhu XX. Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network. IEEE Trans Geosci Remote Sens. 2018;56:6699-711. doi: 10.1109/TGRS.2018.2841808.
19. Benjdira B, Bazi Y, Koubaa A, Ouni K. Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images. Remote Sens. 2019;11:1369. doi: 10.3390/rs11111369.
20. Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, et al. A survey of deep learning-based object detection. IEEE Access. 2019;7:128837-68.
21. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2014; p. 740-55.
22. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137-49. doi:10.1109/TPAMI.2016.2577031.
23. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv. 2018;arXiv:1804.02767.
24. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv. 2020;arXiv:2004.10934.

DOI: 10.60101/jarst.2023.254011

25. Youssouf N. Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4. Heliyon. 2022;8(12):e11792.

26. Farid A, Hussain F, Khan K, Shahzad, Khan U, Mahmood Z. A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments. MDPI Applied Sciences. 2023;13(5):3059. doi: 10.3390/app13053059.

27. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv. 2004;10934.

28. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems. 2015;91-9.

29. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD:Single shot multibox detector. In European Conference on Computer Vision. 2016; 21-37.

30. Isaac-Medina BKS, Poyser M, Organisciak D, Willcocks CG, Breckon TP, Shum HPH. Unmanned aerial vehicle visual detection and tracking using deep neural networks: A Performance benchmark. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11-17 October 2021; p. 1223-32.

31. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and real-time tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25-28 September 2016; p. 3464-8.

32. Wojke N, Bewley A, Paulus D. Simple online and real-time tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17-20 September 2017; p. 3645-9.

33. Wang Z, Zheng L, Liu Y, Li Y, Wang S. Towards real-time multi-object tracking. In Proceedings of the 16th European Conference, Glasgow, UK, 23-28 August 2020; p. 107-22.

34. Lu Z, Rathod V, Votel R, Huang J. RetinaTrack: Online single stage joint detection and tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13-19 June 2020; p. 14656-66.

35. Bochkovskiy A, Wang CY, Mark Liao HY. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv. 2020;10934.

36. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS). 2015; 91-9.

37. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Berg. SSD: Single Shot Multibox detector. ECCV 2016: Computer Vision - ECCV. 2016;21-37.

38. Redmon J, Divvala S, Girshick R, Farhadi, A. You only look once: unified, real-time object detection. In: CVPR. 2016.

39. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS. 2015.

40. Szegedy C, Reed S, Erhan D, Anguelov D. Scalable, high-quality object detection. arXiv preprint v3. 2015;arXiv:1412.1441.

41. Erhan D, Szegedy C, Toshev A, Anguelov D. Scalable object detection using deep neural networks. In: CVPR. 2014.

42. Yurtsever E, Lambert J, Carballo A, Takeda K. A survey of autonomous driving: Common practices and emerging technologies. IEEE Access. 2020;8: 58443-69. doi: 10.1109/ACCESS.2020.2983149.