SinkrOn : Jurnal dan Penelitian Teknik Informatika
Volume 4, Number 2, April 2020
DOI : https://doi.org/10.33395/sinkron.v4i2.10553

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Deep Neural Networks Approach for Monitoring Vehicles on the Highway

Amir Mahmud Husein
Universitas Prima Indonesia
Medan, Indonesia
amirmahmud@unprimdn.ac.id

Christopher
Universitas Prima Indonesia
Medan, Indonesia
Aaronranger0502@gmail.com

Andy Gracia
Universitas Prima Indonesia
Medan, Indonesia
andyramli6@gmail.com

Rio Brandlee
Universitas Prima Indonesia
Medan, Indonesia
bradnlee@gmail.com

Muhammad Haris Hasibuan
Universitas Prima Indonesia
Medan, Indonesia
muhammadharis1602@gmail.com

Submitted: Mar 15, 2020
Accepted : Apr 14, 2020
Published : Aprl 14, 2020

ABSTRACT– Vehicle classification and detection aims to extract certain types of vehicle information from images or videos containing vehicles and is one of the important things in a smart transportation system. However, due to the different size of the vehicle, it became a challenge that directly and interested many researchers . In this paper, we compare YOLOv3's one-stage detection method with MobileNet-SSD for direct vehicle detection on a highway vehicle video dataset specifically recorded using two cellular devices on highway activities in Medan City, producing 42 videos, both methods evaluated based on Mean Average Precision (mAP) where YOLOv3 produces better accuracy of 81.9% compared to MobileNet-SSD at 67.9%, but the size of the resulting video file detection is greater. Mobilenet-SSD performs faster with smaller video output sizes, but it is difficult to detect small objects.

Keywords–Deep Learning, Deep Neural Network, Vehicle Monitoring, YOLOv3, MobileNet-SSD

I. INTRODUCTION

Vehicle classification and detection is one of the important things in intelligent transportation systems with the aim of extracting certain types of vehicle information from images or videos that contain vehicles (Sang et al., 2018). Vehicle detection in traffic lanes is very important as a traffic identification control adaptive, information about the presence of a vehicle in a predetermined detection zone. In general, vehicle detection with the traditional approach uses vehicle motion to separate it from a fixed background image then select a number of images labeled as targets with the sliding-window method, then extract features using the gradient-oriented histogram (HOG) method, or scale invariant transform feature (SIFT), then feature extraction results are applied to classification methods such as support vector machines (SVM) and neural networks, but need high computational complexity and produce very large redundant areas (Yinghua Li, Song, Kang, Du, &Guizani, 2018).

In this decade, a deep learning approach with a convolutional neural network (CNN) based method has been widely applied for vehicle detection for traffic surveillance systems, a region-based and regression-based approach is a two-stage method by producing candidate object boxes through various algorithms then classifying objects by Convolutional neural networks such as R-CNN, Fast R-CNN Spatial Pyramid Pooling Network (SSP-Net) and R-CNN Faster have produced high accuracy by optimizing the Selective Search function to the Regional Proposal Network (RPN), but are too computationally intensive, the bound box is too slow for real-time or close to real-time (Zhang, Li, & Yang, 2019). Single stage detection models such as Single Shot Multibox Detector (SSD) (W. Liu et al., 2016) and You Only Look Once (YOLO) (Redmon, Divvala, Girshick, & Farhadi, 2016) do not produce candidate boxes but directly change the candidate the problem of determining the position of an object bounding box becomes a regression problem for processing (Song, Liang, Li, Dai, & Yun, 2019).

SSD and YOLO algorithms use advanced feed convolution networks to directly predict object classes and locations, which are trained end-to-end. YOLO is a fast algorithm, with relatively low accuracy and effectively overcomes the difficulties caused by changes in the appearance of objects, while the SSD algorithm focuses on detecting objects of different scale with several layers in a ConvNet (Ye, Wang, Song, & Li, 2018), extracting anchors from various aspect ratios and scales on several feature maps (L. Chen, Ye, Ruan, Fan, & Chen, 2018), ignoring the relationships between the various layers of the pyramid of features so that they have relatively poor performance in traffic control small vehicles (Zhang et al., 2019). MobileNet-SSD is one object detection framework using the MobileNet base to extract image features, Network SSD is a classification regression model and bound box regression and output layer for exporting detection results that are proven to produce good and fast accuracy because the MobileNet architecture reduces complexity computing (Yiting Li, Huang, Xie, Yao, & Chen, 2018) (Kevin, Gunawan, Zagoto, Laurentius, & Husein, 2019).

In this paper an in-depth learning method proposed in vehicle detection using YOLOv3 (Redmon & Farhadi, 2018) is an improvement from YOLOv2 (Redmon & Farhadi, 2017) and MobileNet-SSD. Specifically, we recorded highway flow data at the intersection of Medan using the iPhone5 and Asus MaxPro M2 mobile devices, the recording dataset was divided into three parts, recording morning, afternoon and evening, then the detection method was tested according to highway current conditions by comparing the detection results YOLO and MobileNet-SSD methods.

II. RELATED WORK

Detection of vehicles by category inference on video sequence data is an important but challenging task in an urban traffic surveillance system with the primary goal of extracting vehicle features from videos or images captured by traffic surveillance, then identifying vehicle types, and providing reference information for monitoring and traffic control, some researchers propose a deep learning approach using neural networks which in this decade have made progress in vehicle detection, such as (Yang, Li, & Lin, 2018) proposing a multi-perspective convolutional neural network (Multi-PerNet) to extract features Remote visual image of vehicle object detection, the Multi-PerNet Model extracts feature maps, while k-means clustering is used as area distribution and object-area aspect ratio in sample images, Faster-R-CNN framework is applied as object classification and detection models . In the work (H. Wang &Cai, 2014) proposed a deep belief network (DBN) architecture for vehicle classification, (Sang et al., 2018) proposed an increase in YOLOv2 performance in vehicle detection, the k-means ++ grouping algorithm is used to group boxes. vehicle boundaries in the training data set, and six anchor boxes of different sizes were selected, with the aim of reducing the influence of vehicles of different sizes on the detection model, while also increasing the loss function by normalization, while (X. Li, Liu, Zhao, Zhang, & He, 2018) proposed an in-depth learning method for multitarget vehicle detection from Traffic Video with an improved YOLO-vocRV model name network to discuss changes in vehicle appearance based on the YOLOv2 network architecture and (Lestari, Manik, Br Sihotang, & Husein, 2019) proposed a framework YOLOv3 features a network adaptation based on Darknet-53 in the video dataset.

The application of compressed-sensing (CS) theory to produce maps of significance in labeling vehicles in the images presented (Yinghua Li et al., 2018) thereby increasing the classification results of convolutional neural network (CNN) methods, (Nguyen, 2019) made changes to R-CNN Faster architecture, MobileNet was adopted to build the basic convolution layer in the R-CNN Faster, then replace the NMS algorithm with soft-NMS to solve the problem of duplicate proposals, the DP-SSD method (Zhang et al., 2019) was proposed to detect various types vehicles in real-time based on conventional SSD architecture that is upgraded. Modifications to the FasterR-CNN architecture to improve accuracy of vehicle detection are presented in (Fedorov, Nikolskaia, Ivanov, Shepelev, & Minbaleev, 2019), new pose estimation methods based on convex models and put inference are proposed (K. Liu & Wang, 2019) to detect vehicles dynamically quickly and accurately in road scenarios, while (Song et al., 2019) applying road segmentation to offer higher detection accuracy, especially the problem of detecting small vehicle objects on the YOLOv3 architecture. Adaptive Perceive-SSD (AP-SSD) is proposed (X. Wang et al., 2018) based on an improved SSD object detection framework for accuracy and speed of multi-object detection in traffic scenes.

III. Suggested Method

YOLO (You Only Look Once) and SSD (Singe Shot MultiBox Detector) are one-stage object detection algorithms that treat object detection as a simple regression problem by taking input images, studying class probabilities and bounding box coordinates. This section will explain the introduction to the YOLOv3, SSD and MobileNet-SSD models.

III. 1 YOLO Model

YOLO was proposed by Joseph Redmon et al (Redmon et al., 2016) to adopt a different approach from the previous network by not undergoing a regional proposal step such as R-CNN. YOLO implements a single detection network that unites the two detector components; object detectors and class predictors. YOLOv2 (Redmon &Farhadi, 2017) is able to detect more than 9,000 objects, is trained on ImageNet and MS COCO data sets and has reached 16% of the average Precision (mAP) which is not good enough but is very fast during the test time, while YOLOv3 (Redmon &Farhadi, 2018) greater, using network adaptation features based on Darknet-53 as feature extraction and SoftMax loss in YOLOv2 replaced by independent logistic classifiers and binary cross-entropy loss to avoid label overlaps during multilabel classification, the number The anchor box is changed from five to three and uses a network of residual depths to extract image features by applying multi-scale predictions.

The YOLO network divides the input image into a S × S cell grid. If the center of the ground truth box falls into the cell, the cell is responsible for detecting the presence of objects. Each cell predicts (a) the location of B in the bounding box, (b) the trust score, and (c) the probability of the object class being conditioned on the existence of objects in the bounding box. The four coordinate values (tx, ty, tw, th) for each bounding box and the trust score are the output of the input image directly through regression operations, as well as the class probability. The trust score represents the accuracy of the bound box that was predicted when the grid contained objects. In the training phase, three feature maps (13 × 13, 26 × 26, 52 × 52) output from the feature extract network. Taking the 13x13 feature map as an example, the proposed method divides the feature map into a 13x13 grid. Each box is responsible for object detection if basic truths are contained in it. predictions will be obtained as:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{1}$$

where (C_x, C_y) states that the center of an object is detected in a grid offset from the top left corner of the feature map; (p_w, p_h) indicate the width and height of the previous anchor box, respectively; and (t_x, t_y, t_w, t_h) are the four offset coordinates predicted by the network. Using sigmoid to compress t_x and t_y to [0, 1], the center of the target can be effectively confirmed in the cell prediction executing the grid cell.

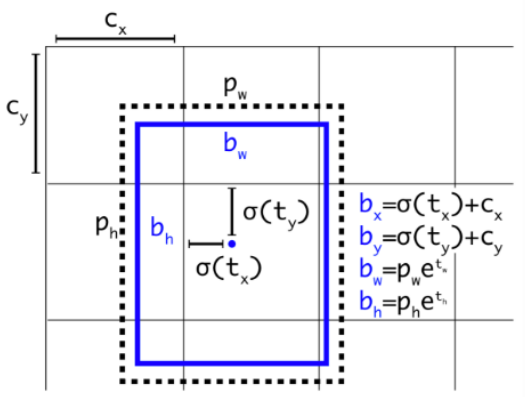


Figure 1. Boundary box with prior dimensions and prediction locations

The YOLOv3 loss function consists of three parts: coordinate prediction errors (conditions 1 and 2), trust scores (conditions 3 and 4) which are intersection crossing errors (IoU), and classification errors. The loss function is defined as follows:

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(\chi_i-\hat{\chi}_i\right)^2+\left(y_i-\hat{y}_i\right)^2$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(\sqrt{w_i}-\sqrt{\widehat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2$$

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2 \qquad (2)$$

$$+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$

$$+\sum_{i=0}^{S^2}1_{ij}^{obj}\sum_{c\epsilon classes}\left(p_i(c)=\hat{p}_i(c)\right)^2$$

where1_{ij}^{obj}indicates that the target was detected by the j boundary box from grid i. To increase the loss of the bounding box coordinate predictions and reduce the loss of confidence predictions for boxes that do not contain objects, the parameters λ_{coord} dan λ_{noobj} are introduced and both are set to 5. Then, $\hat{x}_i,\widehat{y}_i,\widehat{w}_i,\widehat{h}_i$ is a bounding box parameter that is predicted from the center coordinates and size of the box. x_i,y_i,w_i,h_i. $\hat{C}_i$ is the actual parameter, is the prediction of the trust score, C_i is the actual data; $p_i(c)$shows the true value of the object probability on grid i belonging to class C; and $\hat{p}_i(c)$is the predicted value. Except for box size errors, which use average square error, others use binary $l$cross − entropy losses $(a,\hat{a})$ which are defined as follows:

$$l(a,\hat{a})=-a\log\hat{a}_i+(1-a_i)\log(1-\hat{a}_i) \quad (3)$$

III. 2 THE MOBILENET MODEL

The MobileNet model (Howard et al., 2017) is one of the network models developed to improve learning performance in real time with limited hardware conditions without reducing the number of parameters and sacrificing accuracy. The basic convolution structure of the MobileNet network is shown in Figure 2 where Conv_Dw_Pw is a deep and separable convolution structure, consisting of deep layers (DW) and wise layers (Pw). Dw is the convolutional layer that uses 3 × 3 kernels in the kernel, while Pw is the common convolutional layer that uses 1 × 1 kernels. BN is a batch normalization, Conv is a convolution and each convolution result is treated by a batch normalization algorithm and a rectified liner unit (ReLU) activation function).

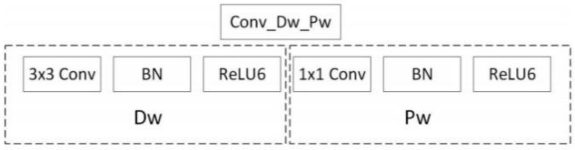


Figure 2. Basic structure of MobileNet convolutional

The ReLU activation function is replaced by ReLU6, and normalization is performed by a batch normalization algorithm (BN), which supports automatic adjustment of data distribution. The ReLU6 activation function can be stated as:

$$y \;=\; min(max(z,0),6) \qquad (4)$$

where z is the value of each pixel in the feature map. Deep and separate convolutional structures allow MobileNet to speed up training and greatly reduce the number of calculations. The standard convolution structure can be stated as:

$$G_N=\sum_M K_{M,N}*F_M \qquad (5)$$

where $K_{M,N}$ is a filter; and M and N are the number of input channels and output channels, respectively. During standard convolution, input images, including feature images, $F_M$means input images, including feature maps, which use the zero padding fill style. When the input image size and channel are respectively $D_F * D_F$dan M, it is necessary to have N filters with M channels and DK size $*$ DK before producing N feature images of size$D_K * D_K$.Computational costs are $\mathrm{D_K * D_K * M * N * D_F * D_F}$.

Instead, the Dw formula can be stated as:

$$\hat{G}_M = \sum \hat{K}_{1,M} * F_M \qquad (6)$$

where $\hat{K}_{M,N}$is the filter. $F_M$has the same meaning as Formula (5). When the step size is one, the zero fill ensures that the size chart characteristics do not change after the application of a deep and separate convolutional structure. When the step size is two, zero filling ensures that the feature graph size is obtained after the application of a deep convolutional structure and can be separated into half of the input image / feature graph; that is, dimensional reduction operations are realized.

The MobileNet v2 network (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) introduces inverted residual and linear bottleneck layers, enabling high accuracy / performance in cellular and embedded vision applications. MobileNet v2 is adapted for object classification and detection, and semantic segmentation. The structure of the standard convolution layer of MobileNet v1 and MobileNet v2 is illustrated in Figure 3.

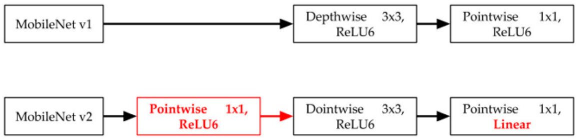


Figure 3. Comparison of convolutional blocks between architectures (L. C. Chen, Sheu, Peng, Wu, & Tseng, 2020)

The MobileNet v2 layer is based on the convolution layer structure of the ResNet architecture which uses shortcuts to improve the accuracy of the deep convolution layer without having a large overhead. The bottleneck layer is used to reduce the size of the input and the upper limit is applied to the ReLU layer which aims to limit the overall complexity.

III.3 Single Shot MultiBox Detector Model

Single Shot MultiBox Detector (SSD) (W. Liu et al., 2016) is one of the first object detection models to use a hierarchy of pyramid features of convolutional neural networks for efficient detection of objects of various sizes. SSD uses the VGG-16 model that was trained before on ImageNet as its basic model for extracting useful image features. The SSD network is a regression model, which uses different convolution layer features to make classification regression and boundary regression. This model resolves the conflict between translation invariance and variability, and achieves good precision and detection speed. The SSD framework is illustrated in figure 4.

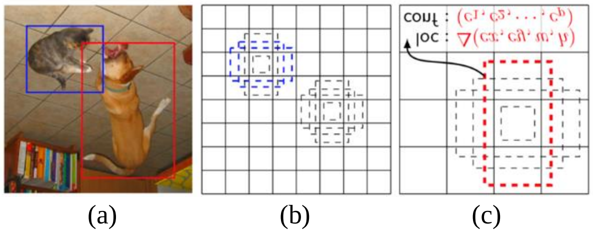


Figure 4. SSD Framework. *(a) Training data contains pictures and basic truth boxes for each object. (b) In the fine-grained feature map (8 x 8), different aspect ratio anchor boxes correspond to smaller areas of raw input. (c) In the coarse grained feature map(4 x 4), the anchor box covers a larger area than the raw input*

The SSD predicts a predetermined anchor box offset (default box) for each feature map location. Each box has a fixed size and position on the cell. All tile anchor boxes throughout the map feature in a convolutional way. Feature maps at different levels have different receptive field sizes. Anchor boxes at different levels are rescaled so that a feature map is only responsible for objects at a certain scale. Figure 5 reveals a default box on the feature map from different convolutional layers

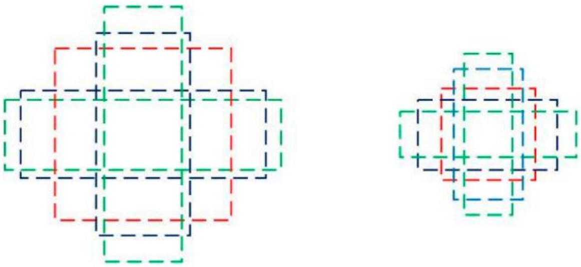


Figure 5. Default box on the feature map

Each default box predicts a class B score and four position parameters. Therefore, B ∗ k ∗ w ∗ h class scores and 4 ∗ k ∗ w ∗ h position parameters must be predicted for image features w ∗ h, requiring a (B + 4) 3 × 3 sized convolution kernel to process the feature map. Then, the convolution results must be taken as the final features for classification regression and boundary box regression. Here, B is set to four because there are four typical defects on the sealing surface of the container in the filling line. The default box scale for each feature map is calculated as:

$$S_K = S_{min} + \frac{S_{max} - S_{min}}{m-1}(k-1), (k \in [1, m]) \qquad (7)$$

where m is the number of feature maps; and S_{max}, S_{min} are parameters that can be set. To control the reasonableness of feature vectors in training and experimental trials, the same five types of width-to-height ratio$a_r = \{1, 2, 3, 0.5, 0.33\}$are used to produce a standard box. Then, each box is calculated the width of the default box $w_k^a = S_k\sqrt{a_r}$ and the height of the default box $h_k^a = S_k/\sqrt{a_r}$. Next, a default box $S_k' = \sqrt{S_k S_{k+1}}$ must be added when the width-to-height ratio is one. The center of each default box becomes $\left(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|}\right)$, and $|f_k|$ is the feature unit size $k, i, j \in [0, |f_k|]$. SSD is an end-to-end training model. The overall loss function of the training contains loss of confidence in $L_{conf}(s, c)$from the classification regression and loss of position from theregression boundary box $L_{loc}(r, l, g)$. This function can be described as:

$$L(s, r, c, l, g) = \frac{1}{N}\left(L_{conf}(s, c) + \alpha L_{loc}(r, l, g)\right) \qquad (8)$$

where α are the parameter for balancing loss of trust and loss of position; s and r are eigenvectors of loss of confidence and loss of position, respectively; c is classification confidence; l is the offset from the prediction box, including the translation of the middle coordinate translation and the height and width scaling offset; g is the calibration box of the target's actual position; and N is the number of default boxes that match the calibration box from this category.

IV. Result and Discussion

In this section, we describe the results of the performance evaluation of the proposed vehicle object detection algorithm, in particular we built a video dataset of traffic flow activity at one of the intersections of Medan City Jl. Kejaksaan, No.5EE 2nd floor, Petisah Tengah, Medan Petisah, for seven days using two mobile devices produced 42 highway videos. The device specifications are shown in Table 1.

Table 1. DEVICE SPECIFICATIONS

Brand\Spesifications	RAM	Internal Memory	Camera Resolution
iPhone 4s	512MB	64GB	8MP/1080p
Asus Zenfone Maxpro M2	6GB	64GB	12MP/2160p(4K)

The process of taking video dataset on the two devices is not perfectly aligned because it has several different resolution, the angle and time of video recording there is a difference - + 3 seconds, besides that the video capacity is stored there are differences between the two devices as shown in Table 2.

Table 2. VIDEO DATASETS

Brand\Spesifications	Total Size(MB)
iPhone 4s	16.251
Asus Zenfone Maxpro M2	10.924

Video recording of the two devices is done at the same time according to the light conditions and traffic density, we collect data according to different traffic flow conditions, namely morning at 08.00-09.00, noon 12.00-01.00 and afternoon at 04.00-05.00, some datasets with traffic density conditions can be shown in Figure 6.

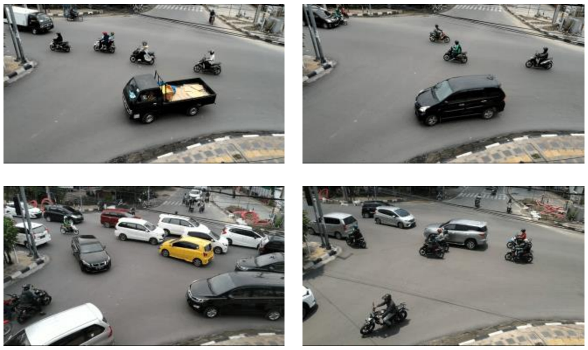
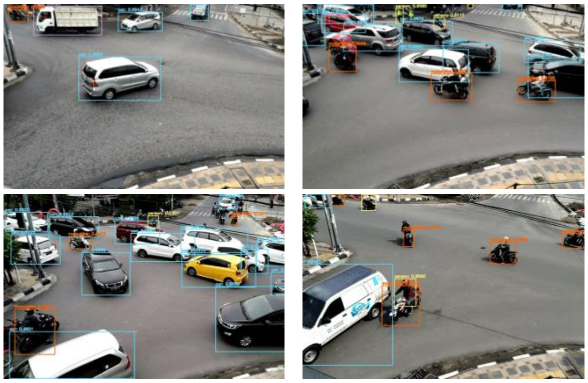
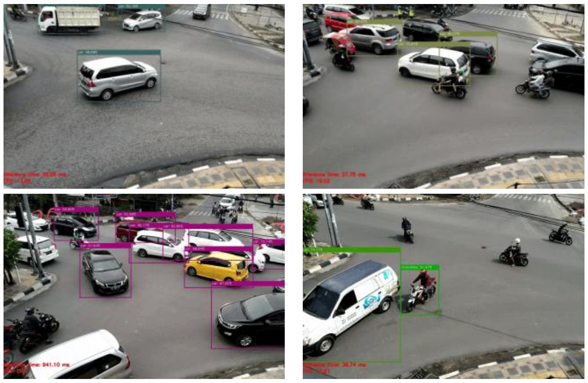


Figure 6. Video Dataset

The testing process of YOLOv3 and MobileNet-SSD detection methods uses hardware with the 7th Generation Intel Core i7 specifications, 8GB DDR4 RAM, GPU GTX1050TI 2GB GDDR5. Three types of vehicles: car, truck and motorbike. First we evaluated the YOLOv3 and MobileNet-SSD models on the dataset taken from the first video device, the dataset contained 18,900 frames, then the second device video dataset of 18,585 frames with a total of 37,758 frames with a frame rate of 25 frames per second in each dataset. YOLO v3 test results on the first device are shown in Figure 7.a and MobileNet-SSD in Figure 7.b and the second device in Figure 8.a for the YOLO v3 and 8.b for Mobilenet-SSD. Testing the two proposed methods for vehicle detection directly on the video dataset using a pre-trained model on the COCO and ImageNet dataset and then processing the video file by initializing and reading the dimensions of the video frame by chance to find the amount of time producing annotated video output.

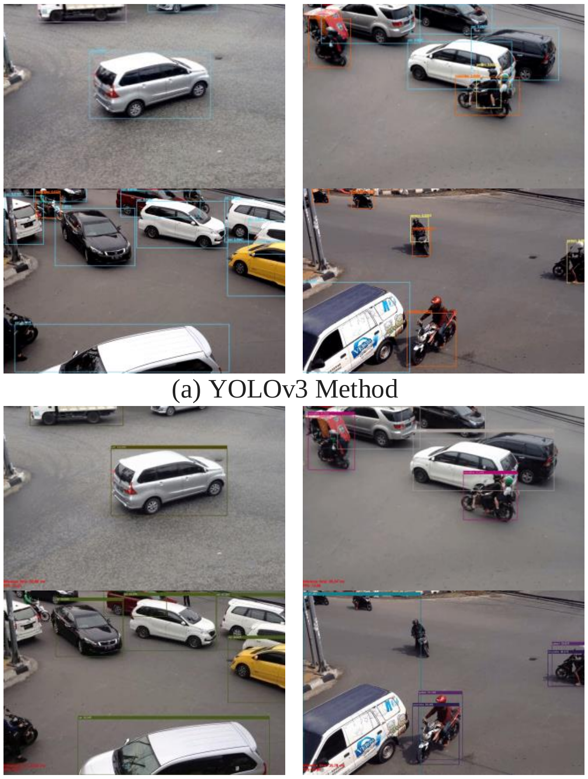


(a) YOLOv3 Method



(b) MobileNet-SSD Method

Figure 7. Detection results on the ASUS dataset (a) YOLO v3, (b) MobileNet-SSD



(a) YOLOv3 Method

(b) Mobilenet-SSD Method

Figure 8. Detection results on the iPhone dataset (a) YOLOv3, (b) MobileNet-SSD

The YOLOv3 approach is difficult to detect adjacent small objects and grouped objects because it divides the input image into an SxS grid where each cell in the grid predicts only one object, while MobileNet-SSD has faster performance, easy to practice by producing better video sizes, but harder with smaller objects.

Table 3. MAP COMPARISON

Models	**mAP**	**car**	**truck**	**motorbike**
YOLOv3	81.9	98.5	87.7	54.9
MobileNet-SSD	67.9	96.7	86.7	43.1

V. CONCLUSION

In this study, we propose a YOLOv3 one-stage detection method by comparing the MobileNet-SSD method for vehicle detection and classification on a highway surveillance video dataset using two mobile devices, testing is done using the original model, the YOLOv3 model approach results in a better accuracy but slower and harder about small objects in groups. MobileNet-SSD has faster performance and produces smaller video file sizes than YOLOv3. The results of our experiments show that both detection methods need to be considered in real-world application, especially the problem of small object detection, vehicle tracking and calculation, this will be our consideration for further work, besides optimizing the algorithm to further improve its effectiveness and efficiency.

VI. REFERENCES

Chen, L. C., Sheu, R. K., Peng, W. Y., Wu, J. H., & Tseng, C. H. (2020). Video-based parking occupancy detection for smart control system. *Applied Sciences (Switzerland)*, *10*(3). https://doi.org/10.3390/app10031079

Chen, L., Ye, F., Ruan, Y., Fan, H., & Chen, Q. (2018). An algorithm for highway vehicle detection based on convolutional neural network. *Eurasip Journal on Image and Video Processing*, *2018*(1), 1–7. https://doi.org/10.1186/s13640-018-0350-2

Fedorov, A., Nikolskaia, K., Ivanov, S., Shepelev, V., & Minbaleev, A. (2019). Traffic flow estimation with data from a video surveillance camera. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0234-z

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Retrieved from http://arxiv.org/abs/1704.04861

Kevin, K., Gunawan, N., Zagoto, M. E. K., Laurentius, L., & Husein, A. M. (2019). Comparison Of Cellular Video Quality For Object Detection Using Neural Network Convolution. *SinkrOn*, *4*(1), 260. https://doi.org/10.33395/sinkron.v4i1.10248

Lestari, P., Manik, D. H. D., Br Sihotang, N. L., & Husein, A. M. (2019). Video Surveillance System with a Deep Learning Approach. *SinkrOn*, *4*(1), 263. https://doi.org/10.33395/sinkron.v4i1.10247

Li, X., Liu, Y., Zhao, Z., Zhang, Y., & He, L. (2018). A Deep Learning Approach of Vehicle Multitarget Detection from Traffic Video. *Journal of Advanced Transportation*, *2018*, 1–11. https://doi.org/10.1155/2018/7075814

Li, Yinghua, Song, B., Kang, X., Du, X., & Guizani, M. (2018). Vehicle-type detection based on compressed sensing and deep learning in vehicular networks. *Sensors (Switzerland)*, *18*(12), 1–15. https://doi.org/10.3390/s18124500

Li, Yiting, Huang, H., Xie, Q., Yao, L., & Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences (Switzerland)*, *8*(9). https://doi.org/10.3390/app8091678

Liu, K., & Wang, J. (2019). Fast dynamic vehicle detection in road scenarios based on pose estimation with convex-hull model. *Sensors (Switzerland)*, *19*(14). https://doi.org/10.3390/s19143136

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9905 LNCS*, 21–37.

https://doi.org/10.1007/978-3-319-46448-0_2

Nguyen, H. (2019). Improving Faster R-CNN Framework for Fast Vehicle Detection. *Mathematical Problems in Engineering, 2019*. https://doi.org/10.1155/2019/3808064

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

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. Retrieved from http://arxiv.org/abs/1804.02767

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. *Sensors (Switzerland)*, *18*(12). https://doi.org/10.3390/s18124272

Song, H., Liang, H., Li, H., Dai, Z., & Yun, X. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, Vol. 11. https://doi.org/10.1186/s12544-019-0390-4

Wang, H., & Cai, Y. (2014). A Multistep Framework for Vision Based Vehicle Detection. *Journal of Applied Mathematics*, *2014*(2). https://doi.org/10.1155/2014/876451

Wang, X., Hua, X., Xiao, F., Li, Y., Hu, X., & Sun, P. (2018). Multi-object detection in traffic scenes based on improved SSD. *Electronics (Switzerland)*, *7*(11). https://doi.org/10.3390/electronics7110302

Yang, C., Li, W., & Lin, Z. (2018). Vehicle Object Detection in Remote Sensing Imagery Based on Multi-Perspective Convolutional Neural Network. *ISPRS International Journal of Geo-Information*, *7*(7), 249. https://doi.org/10.3390/ijgi7070249

Ye, T., Wang, B., Song, P., & Li, J. (2018). Automatic railway traffic object detection system using feature fusion refine neural network under shunting mode. *Sensors (Switzerland)*, *18*(6). https://doi.org/10.3390/s18061916

Zhang, F., Li, C., & Yang, F. (2019). Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. *Sensors (Switzerland)*, Vol. 19. https://doi.org/10.3390/s19030594