Journal of Systems Engineering and Information Technology
Vol. 1 No. 1 (2022) Page 27 – 32 | e-ISSN: 2829-310X
Journal Homepage: www.jurnal.iaii.or.id/index.php/JOSEIT

Comparative Analysis of Deep Learning Models for Vehicle Detection

Rendi Nurcahyo[1*], Mohammad Iqbal[2]
[1]Faculty of Electrical Engineering and Information Technology Gunadarma University Depok, Indonesia
[2]Faculty of Computer Science and Information Technology Gunadarma University Depok, Indonesia

[*]Email corresponding author: rendi608@gmail.com

Abstract

There are many Deep Learning model algorithms for each use case such as Object Detection which has several models, including the ones commonly used, namely Faster R-CNN, SSD (Single Shot Detector), and YOLO (You Only Look Once) version 3, but we need to know the best model for Object Detection especially for vehicle detection which will used for surveillance system. From these models we want to compare which model is the best in a real time process. Each Deep Learning model has its own advantages and disadvantages that affect its performance. Therefore, we must determine which model fits our use case and dataset in order to produce the model that has the best performance. Based on these needs, this paper will make a comparative analysis of the Deep Learning model for Vehicle Detection of these models, namely Faster R-CNN, SSD, and YOLO v3 to see the advantages and disadvantages and which one is the best. The parameters used for comparison are MAP, FPS, Latency which represent whether the model is suitable for real time or not. After comparisons were made, it was concluded that of the three models mentioned, only the YOLO v3 model could be used as real time detection because it had low latency. Its inference time is 60% faster than SSD and 85% faster than Faster R-CNN, furthermore YOLO v3 only carried out a single convolution process, making the process simpler and faster without reducing its accuracy.

Keywords: computer vision, deep learning, neural network, object detection, vehicle detection.

1. Introduction

Deep Learning is a part of Artificial Intelligence and Machine Learning which is the development of multiple layer neural networks to provide precise tasks such as object detection, speech recognition, language translation and others. Deep Learning differs from traditional Machine Learning techniques, because Deep Learning automatically makes representations of data such as images, videos or text without introducing code rules or human domain knowledge. [3]

Deep Learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, machine vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance. [3]

Vehicle detection is a very important component in traffic surveillance and automatic driving. The traditional vehicle detection algorithms such as Gaussian Mixed Model (GMM) has achieved promising achievements. But it is not ideal due to illumination changes, background clutter, occlusion, etc. Vehicle detection is still an important challenge in computer vision [2].

Deep learning has significantly improved the state-of-the-art performance in many fields, including natural language processing, computer vision, and recommender systems. In computer vision, techniques take images or videos as input, process with fine-tuned algorithms and produce useful information for humans. [14]

Object Detection, one of the most basic and challenging problems in computer vision, chooses to find examples of objects from a large number of categories that have been determined in natural images. Deep Learning techniques have emerged as a powerful strategy for studying the representation of features directly from data and have led to extraordinary breakthroughs in the field of generic Object Detection. In this paper we compare three models of Deep Learning for Object Detection which are the best for Vehicle Detection. [7]

A comparative study framework is important to enable people who are interested in applying Deep Learning in their research and / or application to make decisions based on information about which model suits their needs. In the following section, a literature survey of previous work in this field is presented.

Accepted by editor: 16-05-2020 | Final revision: 18-10-2020 | Online publication : 25-03-2022

The results are written based on a logical order to form the models considered in this comparative analysis are: Faster R-CNN [18], Single Shot Detector (SSD) [4], and YOLO v3 [11]. All of these models are same usage for Object Detection and have a top 5 ranking.

1.1. Faster R-CNN

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. RPN and Fast R-CNN are merged into a single network by sharing their convolutional features—using the recently popular terminology of neural networks with "attention" mechanisms, the RPN component tells the unified network where to look [16].

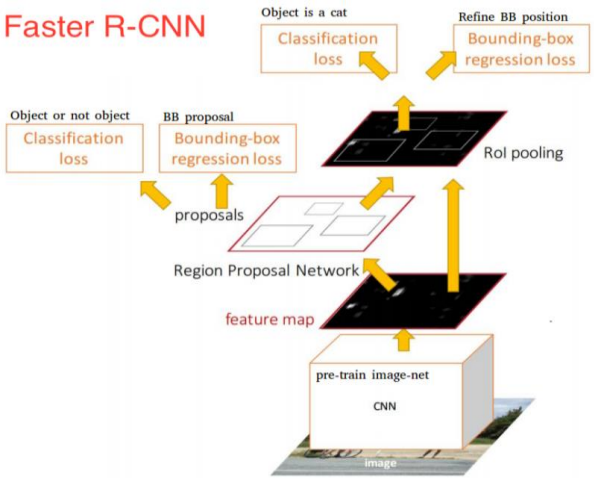

Figure 1. Faster R-CNN architecture [14]

R-CNN is the first step for Faster R-CNN. It uses **s**earch selective to find out the regions of interests and passes them to a Convolution Network. It tries to find out the areas that might be an object by combining similar pixels and textures into several rectangular boxes. The R-CNN paper uses 2,000 proposed areas (rectangular boxes) from search selective. Then, these 2,000 areas are passed to a pre-trained CNN model. Finally, the outputs (feature maps) are passed to a SVM for classification. The regression between predicted bounding boxes and ground-truth bounding boxes are computed. [16]

1.2. Single Shot Detector (SSD)

SSD is designed for object detection in real-time. Faster R-CNN uses a region proposal network to create boundary boxes and utilizes those boxes to classify objects. While it is considered the start-of-the-art in accuracy, the whole process runs at 7 frames per second. Far below what a real-time processing needs. SSD speeds up the process by eliminating the need of the region proposal network. To recover the drop in accuracy, SSD applies a few improvements including multi-scale features and default boxes. These improvements allow SSD to match the Faster R-CNN's accuracy using lower resolution images, which further pushes the speed higher. According to the following comparison, it achieves the real-time processing speed and even beats the accuracy of the Faster R-CNN. (Accuracy is measured as the mean average precision mAP: the precision of the predictions.) [7].

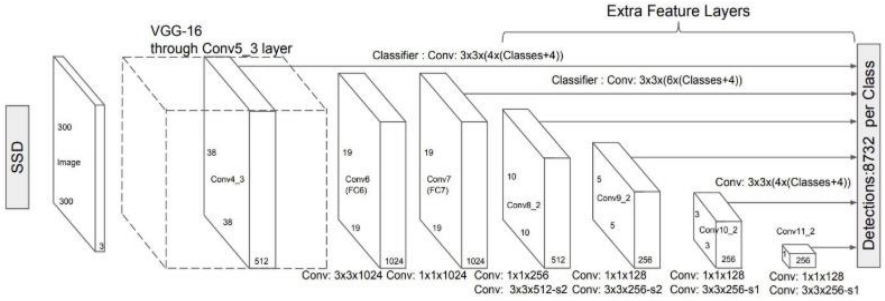

Figure 2. Single Shot Detector (SSD) architecture [7]

SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set of default boxes of different aspect ratios at each location in several feature maps with different scales. For each default box, we predict both the shape offsets and the confidences for all object categories. At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss and confidence loss. [4]

1.3. You Only Look Once (YOLO) version 3

You only look once, or YOLO, is one of the faster object detection algorithms out there. Though it is no longer the most accurate object detection algorithm, it is a very good choice when you need real-time detection, without loss of too much accuracy. Prior work on object detection repurposes classifiers to perform detection. Instead, YOLO frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance [5].

YOLO have some updates to YOLO v3 like design was changed to make it better. It also trained this new network that's pretty swell. It is a little bigger than last time but more accurate and still fast. [6] At 320 x 320 YOLO v3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. It achieves 57:9 AP50 in 51 ms on a Titan X, compared to 57:5 AP50 in 198 ms by RetinaNet, similar performance but 3.8x faster. [11]

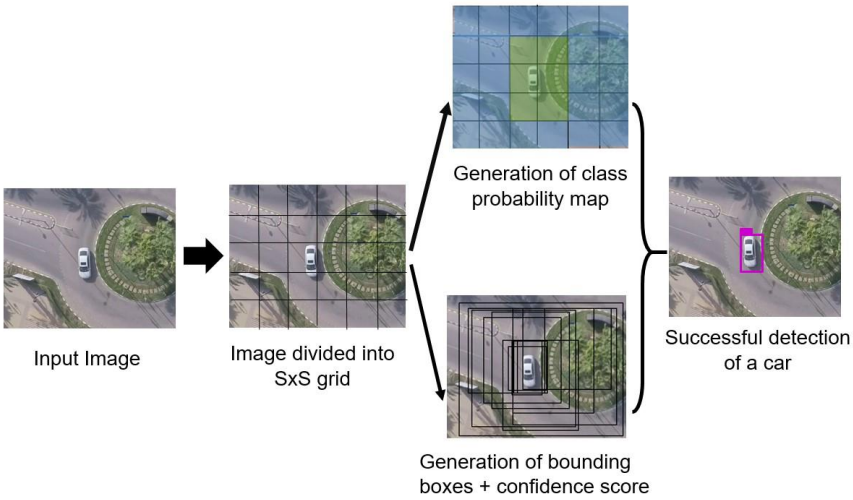

Figure 3. Successive stages of the YOLOv3 model applied on car detection [1]

The newer architecture boasts of residual skip connections, and up sampling. The most salient feature of v3 is that it makes detections at three different scales. YOLO is a fully convolutional network and its eventual output is generated by applying a 1 x 1 kernel on a feature map. In YOLO v3, the detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network [11].

The shape of the detection kernel is 1 x 1 x (B x (5 + C)). Here B is the number of bounding boxes a cell on the feature map can predict, "5" is for the 4 bounding box attributes and one object confidence, and C is the number of classes. In YOLO v3 trained on COCO, B = 3 and C = 80, so the kernel size is 1 x 1 x 255. The feature map produced by this kernel has identical height and width of the previous feature mAP, and it has detection attributes along the depth as described above [11].

Contrary to R-CNN variants, YOLO [30], which is an acronym for You Only Look Once, does not extract region proposals, but processes the complete input image only once using a fully convolutional neural network that predicts the bounding boxes and their corresponding class probabilities, based on the global context of the image. The first version was published in 2016 (Figure 3). Later on in 2017, a second version YOLOv2 [22] was proposed, which introduced batch normalization, a retuning phase for the classifier network, and dimension clusters as anchor boxes for predicting bounding boxes [1].

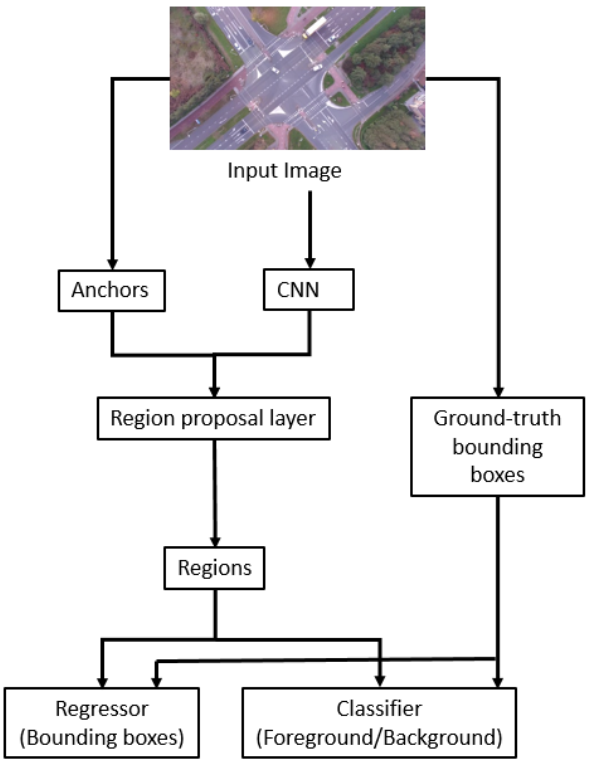

Figure 4. Simple description of YOLO v3 architecture [1]

2. Method

The purpose of this paper is to compare the afore mentioned Deep Learning models (Faster R-CNN, SSD, and YOLO v3) which the best for Vehicle Detection. The models used are pre-trained model which have been trained before using VOC dataset. After that, we set configuration for framework environment and then testing the models in GPU mode with any libraries for computer vision or image processing like OpenCV, CUDA library for NVidia devices, and cuDNN for optimize CUDA library. We are testing models in our environment with different framework, for Faster R-CNN and Single Shot Detector are testing with Tensorflow framework, and for YOLO is testing with itself framework that is Darknet.

Table 1. Configurations of the models

Config	Faster R-CNN	SSD	YOLO v3
Dataset	VOC2007	VOC2007	VOC2007
Input Size	Original image (1024x600)	512	416
Batch	1	32	46
Framework	Tensorflow	Tensorflow	Darknet
Cuda	10.1	10.1	10.1
OpenCV	3.0	3.0	3.0

From the table 1 above, configuration for each model may have different in input size which are to be adjustment by each algorithm.

3. Result and Discussion

The result analysis of a system or algorithm is based upon some set of parameters. Most common parameters are performance, time taken, resource needed, accuracy, etc. which are undertaken in almost all analysis. Where the performance is parameter indicating how well the algorithm does perform. Time taken is the parameter which represents the time taken by the algorithm to output the result. Resources needed are defined as the amount of resources required by the algorithm. Accuracy defined the promising factor of the algorithm which is the percentage of the correct output generated by the algorithm.

On applying general parameters over the Faster R-CNN model, the result show that it is have precision better than YOLO v3 but the speed is too slow because it just can process by 7 FPS and can't processing in real time.

This long process is due to the large number of boxes per region processed, which is around 6000 boxes in neural network.

Similar to Faster R-CNN, Single Shot Detector (SSD) model also has moderate latency because the number of boxes processed per region is 24564 boxes but it makes the SSD model has the highest model precision with a value of mAP 79.8.

And You Only Look Once (YOLO) version 3 is proposed for the recognition of the objects. As the above methods use proposed regions to identify the object in the image, it actually never considers the full image. Rather the regions with high probability of having the objects are passed in the system for the object detection. But in YOLO v3, it has only one Convolutional network and the whole image is analyzed by this network. It divides the image into S x S grid and take m bounding boxes. For each box the network outputs a class probability and the classes with chance higher than the threshold value are used to locate the object. This method has many advantages due to its single convolutional neural network. Firstly, it predicts bounding boxes and class probability directly from the whole image in one evaluation. Secondly, the whole detection process is done in a single network; hence it is easy to optimize the network. It is much faster than the Faster R-CNN and Single Shot Detector (SSD) as it has only one convolutional neural network.

Table 2. Test Result of Different Models

Model	MAP	FPS	Number of Boxes	Latency
Faster R-CNN	73.2	7	~6000	Medium
SSD	79.8	19	24564	Medium
YOLO v3	63.4	46	98	Low

The table 2 shows the test result of different models with respect to latency, mean Average Precision (mAP), Frames Per Second (FPS), and whether they can be used for real time applications or not.

Table 3. Comparison Result of Different Models

Model	Precision	Inference Time	Is Best Model
Faster R-CNN	73.2 %	142 ms	No
SSD	79.8 %	52 ms	No
YOLO v3	63.4 %	21 ms	Yes

The above table clearly shows that YOLO v3 is the best model based on algorithms having low latency and higher FPS. It can be clearly seen that to gain this speed a trade-off has been made in precision. Even after having low mAP, YOLO v3 has acceptable mAP to be able to be used for real time applications and when taken together with the high FPS and latency, it becomes clear it is the best algorithms in its class because speed of detection is the most critical feature for real time application.

4. Conclusion

Speed and accuracy are an important parameter in object detection model for surveillance system which running real time, and YOLO v3 has the best method by simplifying the convolution process to only become a single convolutional process on the neural network, making YOLO v3 successfully reduce the process load, creating high speed but still having good precision.

Future work for YOLO v3 is a deficiency of YOLO v3. This is actually a trick for convolutional process per region with a single process, so it works for objects in large pixel on frames. So, YOLO v3 is not very accurate for detecting objects with a small pixel size, this can be further developed to be a better YOLO model for next version. For vehicle detection, YOLO v3 is still good accurate because the size of the vehicle on the frame is still quite large.

References

[1] Ammar, Adel., Koubaa1, Anis., Ahmed, Mohanned., Saad, Abdulrahman., 2020. *Aerial Images Processing for Car Detection using Convolutional Neural Networks:Comparison between Faster R-CNN and YoloV3*.

[2] Chen, L., Ye, Feiyue., Ruan, Y., Fan, Honghui., Chen, Qimei., 2018. *An Algorithm for Highway Vehicle Detection Based on Convolutional Neural Network*

[3] *Deep Learning*. [wikipedia] (Updated May 5, 2020) Available at : https://en.wikipedia.org/wiki/Deep_learning

[4] Dutta, Suvajit., CS Manideep, Bonthala., Rai, Shalva., V, Vijayarajan.: *A Comparative Study of Deep Learning Models for Medical Image Classification*. (2017)

[5] Hui, Jonathan., *SSD object detection: Single Shot MultiBox Detector for real-time processing*. [Medium] (Updated Mar 14, 2018) Available at: https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

[6] Kathuria, Ayoosh., 2018. *What's new in YOLO v3?* [Towards Data Science] (Updated Apr 23, 2018) Available at: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b
[7] Kazakov, I., 2017. *Vehicle Detection and Tracking*. [Towards Data Science] (Updated May 14, 2017) Available at: https://towardsdatascience.com/vehicle-detection-and-tracking-44b851d70508
[8] Kumar, Prince., Garg, Vaibhav., Somvanshi, Pavan., Pathanjali.: *A Comparative Study of Object Detection Algorithms in A Scene*. (2019)
[9] Liu, Wei., Anguelov, Dragomir., Erhan, Dumitru., Szegedy, Christian., Reed, Scott., Fu1, Cheng-Yang., C. Berg, Alexander.: *SSD: Single Shot MultiBox Detector*. (2016)
[10] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: *You only look once: Unified, real-time object detection. In: CVPR*. (2016)
[11] Redmon, J., Farhadi, Ali.: *YOLOv3: An Incremental Improvement*. (2018)
[12] Ren, Shaoqing., He, Kaiming., Girshick, Ross., Sun, Jian.: *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. (2016)
[13] Shatnawi, Ali., Al-Bdour, Ghadeer., Al-Qurran, Raffi., Al-Ayyoub, Mahmoud.: *A Comparative Study of Open Source Deep Learning Frameworks*. (2018)
[14] Xiao, Y., 2019. *Vehicle Detection in Deep Learning. MSc. Virginia: Virginia Polythecnic Institute and State University*
[15] [15] Xu, Joyce., 2017. *Deep Learning for Object Detection: A Comprehensive Review*. [Towards Data Science] (Updated Sep 12, 2017) Available at: https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9
[16] Xu, Yinghan., *Faster R-CNN (object detection) implemented by Keras for custom data from Google's Open Images Dataset V4*. [Towards Data Science] (Updated Nov 20, 2018) Available at : https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a