The Affect of Image Lighting in Determining Anchor Box for Vehicle Object Detection using Faster R-CNN

Bernardus Hersa Galih Prakoso[1] and *Rosalia Arum* Kumalasanti[2*]

[1,2] Department of Informatics, Sanata Dharma University, Sleman, Yogyakarta, Indonesia

Abstract. Effective traffic management is a critical component in urban safety and the efficiency of road use. Modern traffic system to manage traffic uses the ability to detect traffic to better adjust traffic flow. Object detection can be used in such cases, where CCTV feed is used as a reference to locate and count vehicles. Data collection was carried out during both day and night situations. Faster R-CNN is one of such algorithms that has proven to be robust for object detection. It uses a two-stage process that makes use of anchor boxes to determine the existence of objects in an image. This research aims to find out the effects of differing lighting conditions on the road on the effect of the anchor box used for the model to get the best result. Under both day and night conditions, the best models used anchor size of [64x64; 128x128; 256x256; 512x512] and anchor ratio of [0.5; 1.0; 2.0]. Lighting conditions does not seem to affect the choice of the anchor box set. It is found that smaller and more varied anchor sizes lead to lower RPN loss. While a more diverse set of anchor ratios provides smaller detector loss.

1 Introduction

Proper traffic management is critical to maintaining a working, safe, and efficient network of travel and commerce. Without traffic management, cities would face an ever-increasing travel time, improper road use, possible accidents, and increased pollution from vehicles idling and staying on the road longer [1]. There are a multitude of ways to manage traffic, but at the root of it all is proper real-time vehicle monitoring.

In recent years, there has been a move towards a smarter city, where technology is integrated more and more into how traffic is managed [2]. Object detection is such technology that can be used to aid in vehicle monitoring. Object detection models allow for a more time-efficient way of vehicle monitoring compared to doing it manually and are more easily accessible compared to other methods such as using magnetic road sensors that require road reconstruction or planning on roads that have yet to be built. This is because object detection models require cameras, which are easier to install. Cameras are also sensors that are often already available for manual traffic monitoring.

Among the many object detection models out there, Faster R-CNN is proven to be robust [2]. It is an algorithm that is accurate and fast enough to be used for real-time purposes. It is

* Corresponding author: rosalia.arum@gmail.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

a two-stage algorithm that makes use of a Region Proposal Network (RPN) that generates object proposals based on the existing neural network on a pretrained feature extractor.

In vehicle monitoring, one important aspect that needs to be taken into consideration is the effect of lighting conditions on the accuracy of a detection model. It is understood that different lighting conditions will affect how cameras capture images and videos, so the ability to still optimally detect and classify vehicles at different times of the day with different lighting conditions is important.

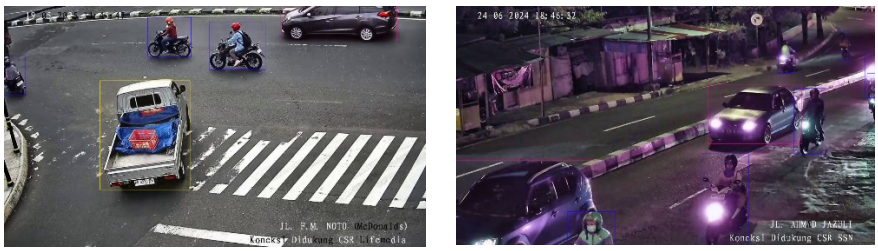
According to this research, choosing the right anchor box dimensions for the RPN stage will have differing effects on the accuracy of a model. The study found that a model with the anchor sizes of 64, 128, 256, 512 and ratios of 0.5, 0.8, 1, 1.2, 1.5, 1.8, 2 obtains the best average precision for their use case [3]. While this specific set of anchor boxes may not work for other use cases, it still showcases the need for choosing the correct set of anchor boxes. While there are studies that show how differently a model can predict an object in different lighting conditions, a study on whether lighting conditions affect which set of anchor boxes performs most optimally has yet to be done.

This study aims to apply the Faster R-CNN model using the ResNet50 network to the features of vehicles and localising them in videos and images using bounding boxes. This paper will focus on determining the optimal anchor sizes and ratios for detections in both day and night conditions. A comparison between the performance of each model in day and night conditions will be done to find out whether different lighting conditions affect average precision.

2 Methods

2.1 Traffic dataset

This research uses four different classes for road vehicles: motorcycles, cars, buses, and trucks. These are the four most common types of vehicles on the road. The dataset is acquired by annotating publicly available road-facing CCTV streams from CCTV Kota Yogyakarta (https://cctv.jogjakota.go.id/) taken during both day and night, where each camera has varying resolutions and qualities. Annotation is done manually to ensure the quality of the annotation. Figure 1 is an example of annotation for images taken during the day (a) and at night (b).



(a) (b)

Fig. 1. Example of annotated image: (a) during the day (b) night.

The resulting dataset consists of 7532 day-images and 8766 night-images. The dataset is split into two splits for training and testing dataset, with a split of 80% for training and 20% for testing. The total number of objects in the dataset is shown in Table 1.

Table 1. Dataset class instances per split.

	Train	Validation	Test
Motorcycles	55211	13674	14611
Cars	24921	8299	8573
Buses	1590	521	540
Trucks	3600	1461	1211

2.2 Augmentation

To compensate for the low amount of data during training, augmentation is used to boost data variance and help in improving model [4]. Augmentation is done during training as each batch of data is used, and each batch will have a random set of augmentations applied to it. The augmentations used consists of affine transformations, brightness and contrast shifts, and channel shuffle done using Albumentations. The un-augmented image is shown in Figure 2.

Fig. 2. Example of un-augmented images.

2.3 Faster R-CNN

In this research, Faster R-CNN based on deep learning method for detecting desired object from a data set. Faster R-CNN is an extension for region convolutional neural network technique [5] [6]. Modelling will be done using Faster R-CNN, which contains three main parts: backbone network, RPN, and detector network. The overall structure of the algorithm is shown in Figure 3.

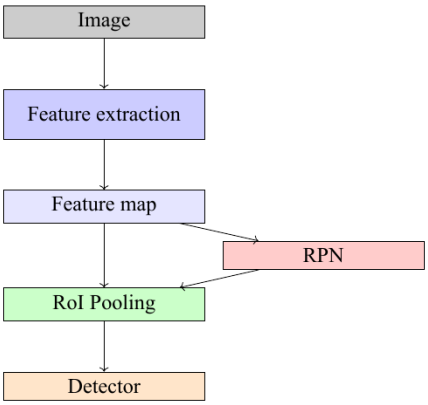


Fig. 3. Structure of Faster R-CNN.

2.4 Backbone network

This paper makes use of ResNet50 as the backbone for the Faster R-CNN model. ResNet50 has been proven to be an accurate and compact classification model. In this case, ResNet50 is to be used for feature extraction. Only the convolutional networks will be used and frozen, as we do not need to train the feature extractor. The rest of the fully connected network that is responsible for classification will be discarded. These implementations are selected on their popularity and performance in the object detection community [4].

2.5 Anchor box tuning

The RPN stage uses a sliding window filter over the feature map produced by the feature extractor, on which anchors of different sizes and ratios are generated [7]. The areas within the bounds of these anchors are then used to determine the object of the sliding window. This is the stage at which the anchor sizes and ratios are tuned during the study. We propose a set of three anchor size sets and three anchor ratio sets, for a total of nine sets of anchor sizes and ratios numbered 0 to 8. The list of anchor sizes can be seen in Table 2 and the anchor ratios can be seen in Table 3.

Table 2. Anchor ratios.

Ratio
0.5, 1.0, 2.0
0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2.0
0.3, 0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2.0, 2.3

Table 3. Anchor sizes.

Size
128*128, 256*256, 512*512
64*64, 128*128, 256*256, 512*512
32*32, 64*64, 128*128, 256*256, 512*512

2.6 Model optimization

As with the original Faster R-CNN paper, this experiment uses Stochastic Gradient Descent (SGD) to optimize model parameters during training. While other research makes use of Adam as an optimizer, SGD is still widely used as it is still very robust. To compensate for the lack of learning rate optimization of SGD during training, Stochastic Gradient Descent with Warm Restarts (SGDR) is used. SGDR is found to be better than using SGD with a static learning rate when it comes to achieving the best training loss [8].

2.7 Model training

We use PyTorch as our framework for building the Faster R-CNN model. A starting learning rate of 0.001 is used and adjusted according to SGDR to a minimum of 0.00001. Learning rate restarts are set to occur every three epochs, which then doubles in length for every cycle. The model is trained for a maximum of 40 epochs, which will stop when there is no improvement in average precision (AP) beyond 1% after 6 epochs. The epoch with the best

AP will be preserved as the final model. Nine models will be trained based on each combination of anchor ratios and sizes.

2.8 Model architecture

Our model uses ResNet50 as the backbone of our feature extraction, where we make use of its existing convolutional layers. The resulting feature map will be fed into our RPN layer, which consists of the aforementioned sliding window filter, making use of anchor boxes [9]. This is would be where our combination of anchor ratios and sizes would be used. Our RPN layer consists of a 3x3 convolutional map, which uses the feature map from ResNet50 as the input. The resulting feature map will then be put into two different layers: a 1x1 convolutional layer meant to classify the object-ness of the area, and a 1x1 convolutional layer meant to regress the bounding box. Non-maximum suppression with a threshold of 0.7 is used in order to reduce redundancy and remove low confidence results.

The region proposals generated by the RPN layer is fed into a Fast R-CNN layer which is used in order to further regress the bounding boxes and classify objects into its corresponding classes.

3 Results and discussion

This research uses image data in the form of traffic CCTV videos that can be accessed through the regional-based CCTV website of the Yogyakarta City Government. There are four categories of vehicles detected in the study: motorcycles, cars, trucks, and buses. The next step is data annotation to provide bounding boxes for the objects. Export annotation data from CVAT and proceed with splitting the data for training and testing. This research also analyses the modelling of the convolutional layer. During this modelling phase, parameters will be adjusted to obtain optimal results. Once the dataset is obtained, it will be processed according to the research flow in Figure 4.

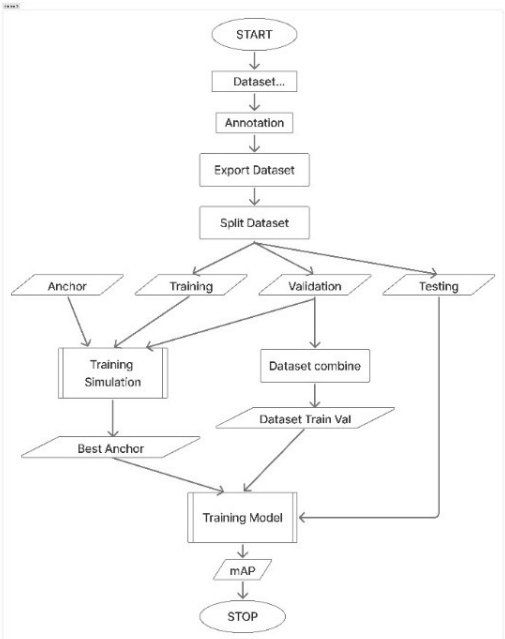


Fig. 4. Flow diagram

3.1 Annotation and export dataset

Annotation is the process of labelling images by marking the positions of objects to be detected with bounding boxes. The annotation process will provide bounding boxes for vehicles that fall into the categories of motorcycles, cars, trucks, and busses. Videos are annotated using the CVAT (Computer Vision Annotation Tool). Export Dataset, the video that has been annotated thru CVAT is exported as a dataset in PASCAL VOC (Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes) format.

3.2 Convolutional layer

In the convolutional layer, feature extraction is performed on the input image. A single image has three color channel : red, green, and blue. Here's an example of performing a convolutional operation on an input with dimensions 5x6x3, where the height is 5 units, the width is 6 units, and the color depth is 3 units. The activation matrix is fed into max pooling with a kernel size of 2x2 and a stride of 1. In order to be used by the next layer, the output of max pooling needs to be converted into a one-dimensional vector. This process is done by flattening the layers. The flatten layer receives a 3x4 input. After that, the flattened vectors and layers need to pass through dense layers with varying weights. This research will compare the influence of anchor size and anchor ratio on model performance during the day and night. The anchor sizes used include [128x128; 256x256; 512x512], [64x64; 128x128; 256x256; 512x512], [32x32; 64x64; 128x128; 256x256; 512x512], and the anchor ratios used include [0.5; 1.0; 2.0], [0.5; 0.8; 1.0; 1.2; 1.5; 1.8; 2.0], [0.3; 0.5; 0.8; 1.0; 1.2; 1.8; 2.0; 2.3].

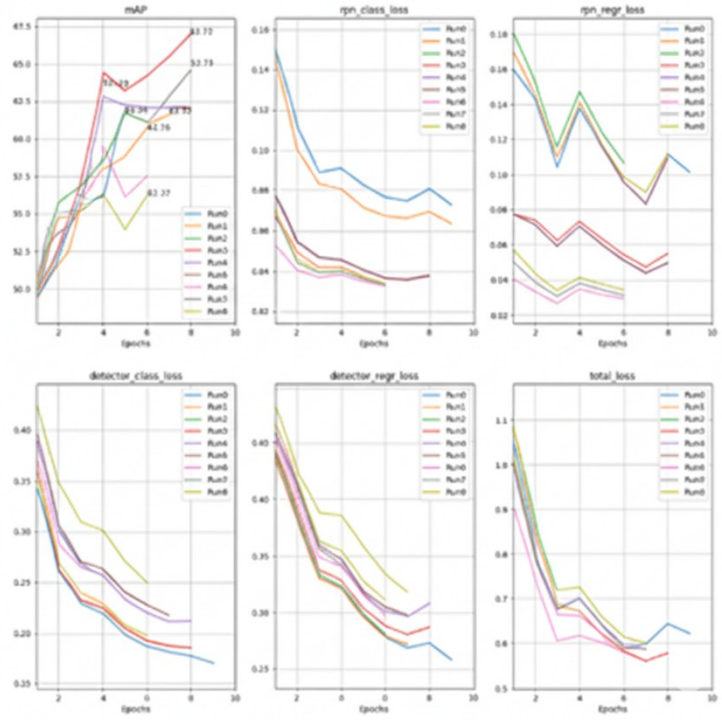


Fig. 5. Run comparison (day).

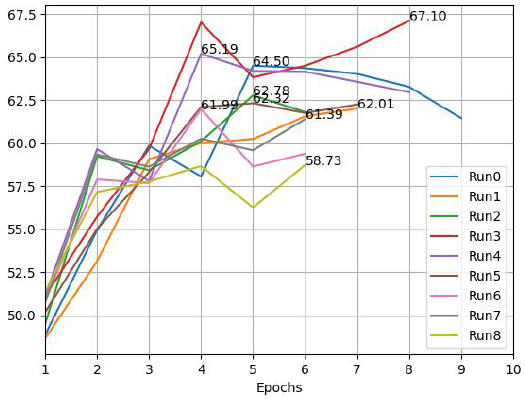


Fig. 6. mAP comparison (day).

Each dataset was used to train nine models with different anchor sizes and anchor ratios. Each model will be named based on the combination of anchor size and anchor ratio. Each model will be named Run0-Run8 based on the combination of anchor size and anchor ratio. The results of the training experiment and mAP can be seen in Figure 5 and Figure 6. The next stage is testing and mAP, using the same combination, which can be seen in Figure 7 and Figure 8.

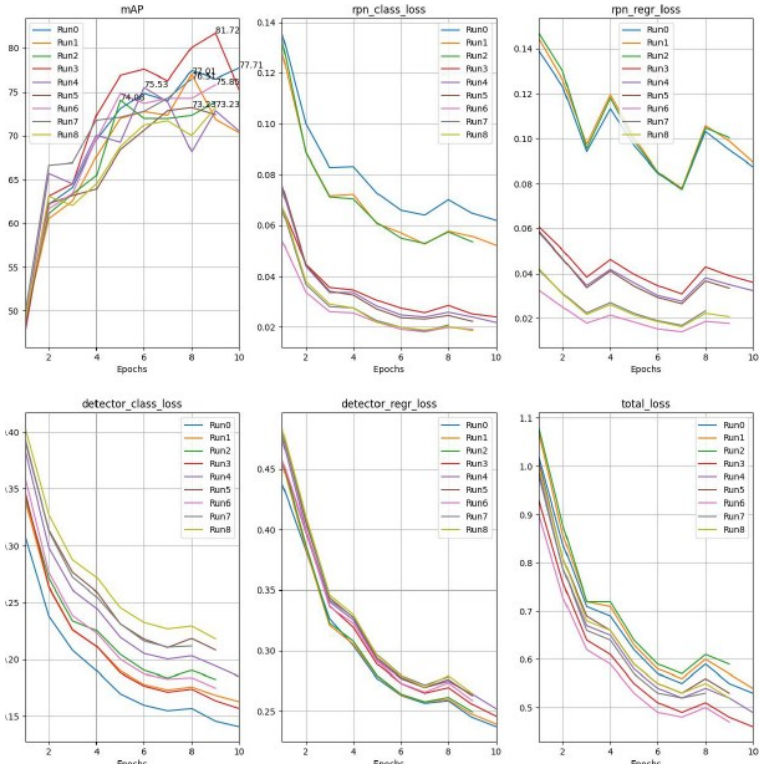


Fig. 7. Run comparison (night).

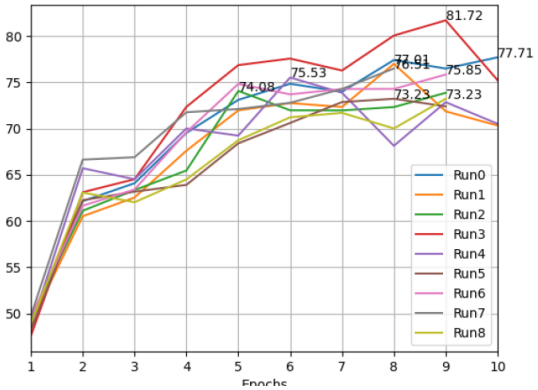


Fig. 8. mAP comparison (night).

4 Conclusion

Based on the results of research conducted on day and night datasets with different anchor sizes and anchor ratios, several conclusions can be drawn. The best model for both day and night conditions is [64x64; 128x128; 256x256; 512x512] and anchor ratio [0.5; 1.0; 2.0]. Lighting does not affect the determination of anchor size and anchor ratio because the models for both day and night dataset achieve maximum mAP using the same anchor boxes. The determination of anchor size and anchor ratio has different effects at each stage of modelling. At the RPN stage, models with more anchor sizes and smaller anchor size tend to have lower loss at that stage, but at the detector stage, models with a slimmer and more granular anchor ration have lower detector loss. The experiment proves that data collection during day or night does not affect anchor boxes.

References

1. J. L. Levy, J. J. Buonocore, K. V. Stackelberg, Evaluation of the public health impacts of traffic congestion: a health risk assessment. Environ Health. vol. **9**, pp. 9-65 (2010). https://doi.org/10.1186/1476-069X-9-65
2. M. T. Audina, F. Utaminingrum, D. Syauqi. Sistem deteksi dan klasifikasi jenis kendaraan berbasis citra dengan menggunakan metode Faster-RCNN pada raspberry Pi 4B. J-PTIIK. vol. **5**, no. 2, pp. 814-819 (2021).
3. J. Fan, T. Huo, X. Li, T. Qu, B. Gao, H. Chen, Covered vehicle detection in autonomous driving based on faster RCNN, in 2020 **39**th Chinese Control Conference (CCC), Shenyang, China, (2020). https://doi.org/10.23919/CCC50068.2020.9189180
4. D. D. Aboyomi, C. Daniel, A comparative analysis of modern object detection algorithms : YOLO vs SSD vs Faster R-CNN. ITEJ. vol. **8**, no. 2, pp. 96-106 (2023). https://doi.org/10.24235/itej.v8i2.123
5. S. M. Sri, B. R. Naik, K. J. Sankar, Object detection based on faster R-CNN. IJEAT. vol. **10**, no. 3, pp. 72-76 (2021). https://doi.org/10.35940/ijeat.C2186.0210321
6. M. D. Yudha, W. Setiawan, Y. Wihardi, Deteksi sepeda motor di jalan raya menggunakan faster R-CNN berbasis VGG16. JATIKOM. vol. **4**, no. 2, pp. 10-13 (2021).
7. F. Joiya, Object detection: YOLO vs Faster R-CNN. International Research Journal of Modernization in Engineering Technology. vol. **4**, no. 9, pp. 1911-1915 (2022).

8. V. Wiley and T. Lucas, Computer vision and image processing: a paper review. IJAIR. vol. **2**, no. 1, pp. 28-36 (2018). https://doi.org/10.29099/ijair.v2i1.42
9. S. Megawan,W. S. Lestari, Deteksi spoofing wajah menggunakan Faster R-CNN dengan arsitektur Resnet50 pada video. JNTETI. vol. **9**, no. 3, pp. 261-267 (2020). https://doi.org/10.22146/.v9i3.231