JURNAL UPI

Journal of Computers for Society

Journal homepage: https://ejournal.upi.edu/index.php/JCS

Detection of Motorcycles on Highways Using Faster R-CNN Based on VGG16

Moch Dian Lazuardi Yudha, Wawan Setiawanb, Yaya Wihardic*

Pendidikan Ilmu Komputer, Universitas Pendidikan Indonesia, Bandung, Indonesia
*Correspondence: E-mail: adian_0856@student.upi.edu

ABSTRACT

This research aims to develop an object detection system with input in the form of image data in free size. The development of an object detection system model was carried out by applying Machine Learning to overcome object detection in an image using the Faster R-CNN method based on the VGG16 algorithm. The system developed produces a bounding box for an object in the image. System development was carried out in the Python programming language by utilizing several libraries such as Keras. Experiments were carried out by measuring the loss value of the training data entered into the system. The experimental results show that the resulting information is proven to be able to detect objects in a given image. This system can produce information based on image data that has been trained with this system. This study used two experiments which obtained a loss value of 0.0601 in the first study and 0.1211 in the second study.

ARTICLE INFO

Article History:
Submitted/Received 14 Jan 2023
First Revised 17 Mar 2023
Accepted 08 May 2023
First Available Online 09 May 2023
Publication Date 15 Jun 2023

Keyword:
Computer vision,
Convolutional neural networks,
Faster RCNN,
Machine learning,
Regional proposal network,
ROI pooling.
VGG16,

1. INTRODUCTION

Vehicles worldwide are becoming more modern and their numbers are increasing with advanced developments (Zhu *et al*., 2014). Vehicles can be detected through a system that takes data from CCTV. This system can utilize the Convolutional Neural Network algorithm which has been bundled with the Keras library and has been proven to be good in terms of performance (Bianco *et al*., 2017). Convolutional Neural Network is an Artificial Neural Network that has many layers and each layer has a two-dimensional plane and each plane consists of several independent neurons (Zhan *et al*., 2016). Generally, Convolutional Neural Networks are used to classify images which will later produce class labels and this process is called machine learning (Wang *et al*., 2017).

The latest machine learning is divided into 2 parts, namely supervised learning and unsupervised learning (Osisanwo *et al*., 2017). Machine Learning is a part of computer science that can teach computers to have the ability to learn without having to be explicitly programmed (Samuel, 1959). Convolutional Neural Network has an architecture called VGG16 which has been proven to produce accurate objects (Song *et al*., 2019). Developed with a more complex architecture, VGG16 is the main basis for the Faster R-CNN algorithm which can produce 91.4% correct objects in a dataset (Song *et al*., 2019). The Faster R-CNN architecture has three main parts (Convolutional Neural Network model, Region Proposal Network, and ROI Pooling Layer) [8]. The Convolutional Neural Network model section can be filled with various Convolutional Neural Network architectures such as resnet and vgg (Deng *et al*., 2018).

The Faster R-CNN architecture based on the VGG16 algorithm requires a GPU with a minimum capacity of 3GB for the training stage, while the YoloV3 architecture requires a GPU with a minimum capacity of 4GB (Song *et al*., 2019). In developing a motorbike detection system in traffic, this research uses the Faster RCNN architecture to study data in the form of images consisting of 600 data. The author also uses libraries in the Python programming language such as Keras as the backend of the system to be developed. The development of a detection system using the Faster RCNN architecture has been built on previous research, including a kiwi fruit detection system which is able to detect kiwi fruit with the help of the Faster RCNN architecture based on the VGG16 architecture (Song *et al*., 2019). There is also other research that can track humans with the Faster RCNN architecture based on VGG16 (Chahyati *et al*., 2017).

2. MOTORCYCLE DETECTION ALGORITHM SYSTEM MODEL ON THE HIGHWAY

VGG16-based Faster R-CNN is divided into four main parts, namely the basic VGG16 algorithm model, Region Proposal Network, ROI Pooling Layer, and Fully Connected Layer (Deng *et al*., 2018). In this research, a configuration was carried out by entering a learning rate of 1e – 5 and 35 epochs. There were no changes to the algorithm architecture in this research. This architectural model depicts a summary of the research development stages (see **Figure 1**).

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxx
e- ISSN 2723-4088

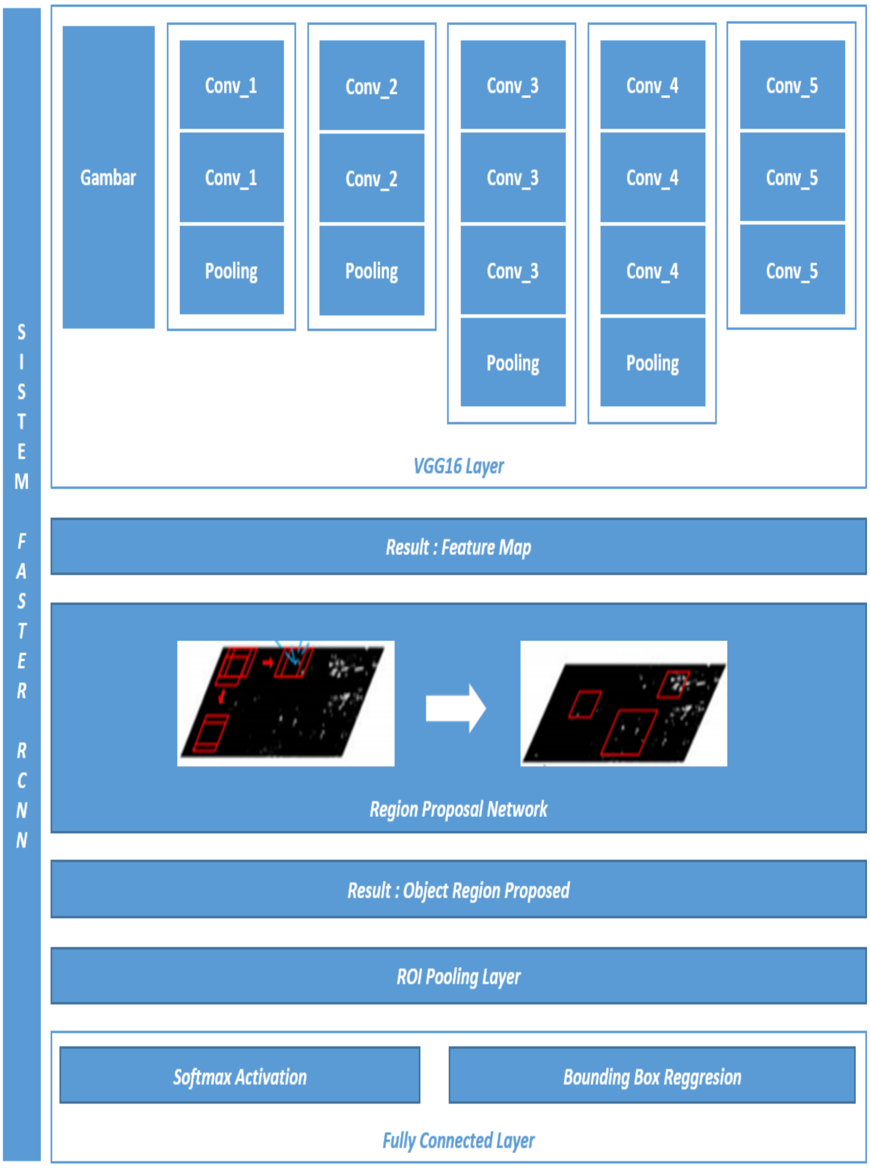

Figure 1. Faster Rcnn algorithm model based on VGG16.

2.1. VGG16 Model Layer

The data that can be processed by this stage is 600 x 800 Pixels. Therefore, you have to resize the image you get. This stage aims to train data that has gone through pre-processing (resizing, labeling objects). The result of this stage is a feature map that will be used for the region proposed network stage.

2.2. Region Proposed Network

In this process, more accurate object retrieval is carried out. To generate region proposals, a sliding window process is carried out using anchors. For the VGG16 model, the anchor dimensions are 512 dimensions (Ren *et al*., 2016). With this sliding window, the classifier determines the probability that a proposal has a real object and regresses the coordinates of the proposal. The anchors used in the Region Proposal Network in this research have scales of 128, 256, and 512 which function to proportion the regions of an object in the image. The result of this stage is the loss of unused bounding boxes on an object because the object's original bounding box has been determined in an image. In the calculation, bounding boxes that have a value of less than 0.7 will later be deleted.

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxxx
e- ISSN 2723-4088

2.3. ROI Pooling Layer

Input from the Region Proposal Network results will be processed by this stage. This stage uses a max pooling layer to convert features in a predetermined region into a smaller feature map or fix the feature map (Girshick, 2015). This stage works by dividing the h x w roi window into an H x W grid of sub windows and then max pooling the values in each sub window into the grid cells (Girshick, 2015).

2.4. Fully Connected Layer

This stage is the final stage of the Faster RCNN architectural model. In this stage there are only softmax and bounding box regression. In the tensorflow library, the author uses softmax and linear activation. This stage aims to predict class objects and bounding box regression.

3. EXPERIMENTAL DESIGN

The experiment was carried out by differentiating two data sets totaling 280 motor objects and 620 motor objects. Both test cases have the same source, namely CCTV Online Dishub Kab. Sukoharjo with video shooting in the morning and afternoon. This research took pictures at this time because the light was suitable for data collection and training by the system. This research uses the Python programming language which is assisted by the Keras library as a backend system, OpenCV as image processing, and Labeling as an application for labeling objects at an early stage. The Faster RCNN algorithm based on VGG16 was created with the help of the Keras library. **Table 1** shows the experimental test cases carried out in this research.

Table 1. Test-case eksperimen.

Experiment	Number of Objects	Number of Epochs	Source
Eks1	280	37	Sukoharjo Transportation Department CCTV website
Eks2	620	35	Sukoharjo Transportation Department CCTV website

4. RESULTS AND DISCUSSION

4.1. Comparison of System Output with Related Research

To clarify how accurate the Faster RCNN algorithm is, the author compares the output with related research such as **Figures 2**, **3 and 4**, we use different data from the two journals. This media was developed using Luther-Sutopo's MDLC method which consists of six stages. Based on **Figure 2**, it can be seen that kiwi fruit can be detected very clearly by the Faster RCNN algorithm. In this research, the author used a dataset of 28,800 kiwi fruit objects (Song *et al.*, 2019). In **Figure 2**, there is a detection error which is marked with a yellow circle. In this system, the yellow circle is a kiwi fruit that was not detected by the system. **Figure 3** depicts human tracking and it can be concluded that Faster RCNN can detect humans on CCTV. **Figure 4** is the result of this research. Researchers only wanted to detect motorbikes facing down, but the system made an error by detecting motorbikes facing up. In this research, the author used 620 motorbike objects for the training stage.

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxx
e- ISSN 2723-4088

Figure 2. Kiwi fruit yield in the morning (Zhan *et al*., 2016).

Figure 3. Results of human tracking with CCTV (Wang *et al*., 2017).

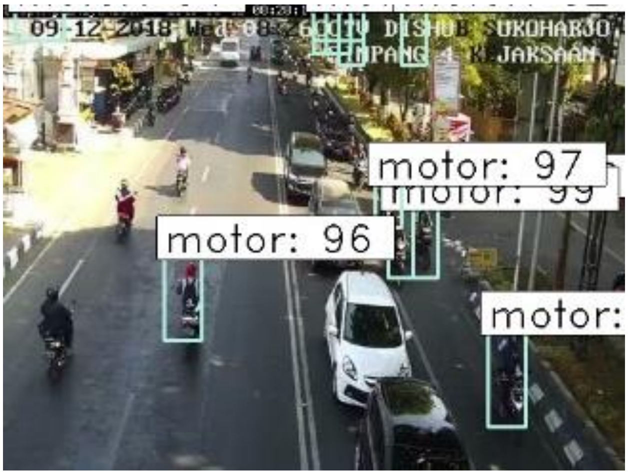

Figure 4. Motor detection results in the system.

4.2. Experiment Results

In this research, 2 experiments were carried out with different amounts of data by assessing the output loss value at each epoch obtained. Therefore, the results of the two experiments are obtained in **Table 2**. Apart from the loss value, the system can detect objects in an image as seen in **Figures 5** and **6**.

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxxx
e- ISSN 2723-4088

Table 2. Readability aspect measurement results.

Epoch	Experiment 1 (280 Objects)	Experiment 2 (620 Objects)
1	2.0636	2.9509
2	0.6355	1.0397
3	0.3967	0.6937
4	0.3227	0.5109
5	0.2881	0.4627
6	0.2262	0.3986
7	0.2053	0.3736
8	0.2405	0.3651
9	0.2683	0.3753
10	0.1695	0.2724
11	0.1388	0.2563
12	0.2088	0.2593
13	0.1463	0.2548
14	0.1481	0.2774
15	0.1732	0.2163
16	0.1348	0.2074
17	0.12	0.1916
18	0.1254	0.2225
19	0.1346	0.1901
20	0.0928	0.1654
21	0.1112	0.1586
22	0.1097	0.1776
23	0.087	0.1702
24	0.0907	0.1759
25	0.1218	0.1528
26	0.113	0.1422
27	0.0842	0.1521
28	0.0856	0.1423
29	0.0737	0.1691
30	0.0786	0.1311
31	0.0682	0.1355
32	0.0844	0.1324
33	0.0658	0.1211
34	0.0601	0.1307
35	0.0796	0.1211
36	0.0703	-
37	0.0627	-

Figure 5. Motor detection results in experiments 1.

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxx
e- ISSN 2723-4088

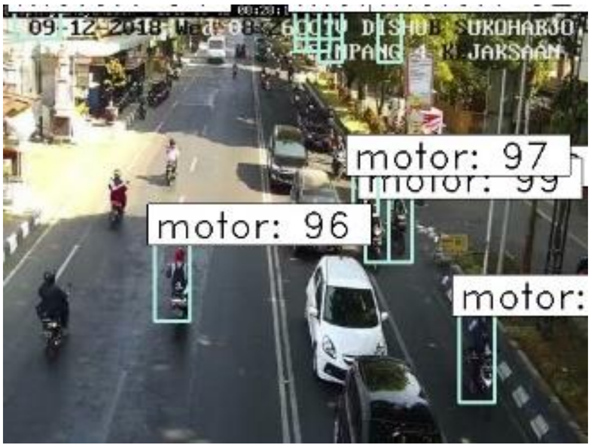

Figure 6. Motor detection results in experiment 2.

4. CONCLUSION

Creating a motorbike detection system in traffic using the Faster RCNN algorithm can be said to be accurate. This system can detect the motorbike that the author wants and can run with different-sized bounding boxes for each object. This algorithm can be developed for other purposes such as detecting traffic violations using CCTV, it can also be used for tracking objects on video or CCTV. However, in this research, the author only used images for the testing stage.

This research answers that the Faster RCNN algorithm can still detect an object even with a relatively small dataset and only 35 epochs in iteration. However, if we look at the experimental results, a dataset that has more data will detect objects more accurately than a smaller dataset.

The benefit of this research is the author can find out how convolutional neural networks can work well with more complex architectures. Currently convolutional neural networks with fast RCNN architecture are widely used in the field of computer vision.

The conclusion from the overall results of the experiments carried out is that the output from the system is not completely accurate, but 95% of the objects detected are accurate. The loss value used in this study was 0.0601 for the first experiment and 0.1211 for the second study. The second experiment also obtained a loss value that was greater than the first value, but the objects detected were more accurate than the first experiment. Most likely because the data used in the second experiment is larger, the system will learn more deeply about the object it wants to detect.

For further research, a system can be developed to detect vehicles that have accidents and vehicles that violate traffic. Apart from that, faster RCNN can also change the basic algorithm model such as using the basic resnet algorithm.

5. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

6. REFERENCES

Bianco, S., Buzzelli, M., Mazzini, D., and Schettini, R. (2017). Deep learning for logo recognition. *Neurocomputing*, *245*, 23-30.

Chahyati, D., Fanany, M. I., and Arymurthy, A. M. (2017). Tracking people by detection using CNN features. *Procedia computer science*, *124*, 167-172.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., and Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, *145*, 3-22.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*, *2015*, 1440-1448.

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., and Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), 1137-1149.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210-229.

Song, Z., Fu, L., Wu, J., Liu, Z., Li, R., and Cui, Y. (2019). Kiwifruit detection in field images using Faster R-CNN with VGG16. *IFAC-PapersOnLine*, *52*(30), 76-81.

Wang, Y., Luo, Z., and Jodoin, P. M. (2017). Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, *96*, 66-75.

Zhan, S., Tao, Q. Q., and Li, X. H. (2016). Face detection using representation learning. *Neurocomputing*, *187*, 19-26.

Zhu, W., Miao, J., Hu, J., and Qing, L. (2014). Vehicle detection in driving simulation using extreme learning machine. *Neurocomputing*, *128*, 160-165.

DOI: https://doi.org/10.17509/jcs.v5i2.xxxxx
e- ISSN 2723-4088