Journal of Integrated and Advanced Engineering (JIAE)
Vol. 3, No. 1, March 2023: 1-6
http://asasijournal.id/index.php/jiae
http://doi.org/10.51662/jiae.v3i1.68

Counting Various Vehicles using YOLOv4 and DeepSORT

Alfan Pahreza Kusumah*, Dena Djayusman, Galih Rizki Setiadi, Ade Chandra Nugraha, Priyanto Hidayatullah
Department of Computer Engineering and Informatics, Politeknik Negeri Bandung, Indonesia

Abstract
The Ministry of Public Works and Public Housing (PUPR) conducted a traffic survey to determine the total number of vehicles and classify them according to the Bina Marga vehicle categorisation. The survey has thus far been carried outmanually. As a result, surveys take a lot of time and money to perform. Additionally, as the survey scope grows, so will the requirement for surveyors. Therefore, a substitute that can execute the survey procedure automatically and with tolerable accuracy is required. One solution is to utilise deep learning technology to detect and categorise vehicles that can be used in apps. The program is designed as a web application that provides a summary of vehicle calculations and receives video data from traffic recordings. The deep learning model used is YOLOv4 which is trained to recognise vehicle classes following Bina Marga vehicle types. The model was trained and tested using the Python programming language and the Darknet framework on the Google Colab platform. The YOLOv4 and DeepSORT method with custom dataset reached a decent accuracy of 67.94%, considering the limited 1000 images used for training the model.

Keywords:
Deep Learning;
Vehicle Detection and Classification;
YOLOv4;

Article History:
Received: July 15, 2022
Revised: March 2, 2023
Accepted: March 9, 2023
Published: March 25, 2023

Corresponding Author:
Alfan Pahreza Kusumah
Department of Computer Engineering and Informatics, Bandung State Polytechnic, Indonesia
Email:
alfan.pahresz.tif91@polban.ac.id

Copyright ©2023 ASASI
This is an open access article under the CC BY-SA license

INTRODUCTION

At the Ministry of Public Works and Public Housing *(PUPR*), a traffic survey was conducted by calculating the total number of vehicles and their classification based on the Bina Marga vehicle class. The Ministry of PUPR, in policy planning and report generation, uses results from traffic surveys. So far, the process of calculating and classifying vehicles in traffic surveys has been done manually by surveyorsby observing one vehicle lane on-site or via Closed-circuit Television (CCTV) camera recordings [1][2]. Given the time and money invested in each survey for calculation and classification and for creating a traffic survey report document, the manual traffic survey process is costly. In addition, there is a high possibility of errors in the calculation and classification of vehicles made by the surveyors because multiple vehicle classes need to be identified and counted simultaneously. Table 1 shows the vehicle classes, [3], that are used as detection classes in this study [4, 5, 6].

This study aims to propose an alternative method of traffic survey that could potentially reduce the dependency on human surveyors using deep learning concepts such as object detection, classification, and tracking. A website-based application that performs object identification, classification, and calculation was proposed to help analyse the YOLOv4 and DeepSORT [7][8] method results with the dataset from CCTV placed on the highway.

Table 1. Bina Marga Vehicle Classes

Classes	Class Name	Vehicle Examples
1	sepeda_motor	Motorcycle, scooter, and another three-wheeled motorised vehicle
2	sedan_jeep_wagon	Sedan, jeep, and Station Wagon
3	angkutan_sedang	Oplet, oplet pick-up, suburban, combi, and minibus
4	pick-up_mikro_truk	Pick-up, micro truck, delivery car or box pick-up
5a	bus_kecil	Small Bus
5b	bus_besar	Large Bus
6a	truk_2_sumbu_4_roda	2-axle 4-wheeled truck
6b	truk_2_sumbu_6_roda	2-axle 6-wheeled truck
7a	truk_3_sumbu	3-axle truck
7b	truk_gandeng	trailer truck
7c	truk_semitrailer	semitrailer truck
8	tidak_bermotor	non-motorised vehicle

YOLOv4 is used despite the existence of YOLOv5 because of better performance characteristics and relatively lower training time. Whereas DeepSORT is chosen because it has an average Multiple Object Tracking Accuracy (MOTA) of 68.7% and provides a better detection and tracking speed [9, 10, 11, 12].

The web application will be able to accept video files and upload them through an API to be processed by the server in Google Colab. The server would then start the detection, classification, tracking and counting onthe video using YOLOv4. The YOLOv4 model must first be converted after training and used in the Tensorflow framework for the DeepSORT algorithm to work. A survey document with road data and vehicle calculation results will be generated using the calculation results with a specifically formatted excel file [10].

METHOD

The research is conducted by building an application to count the number of vehicles by type in a determinedamount of intervals. A custom-trained deep learning model is required to be able to detect custom vehicle classes to do that. Therefore, the YOLOv4 model will be trained to detect 12 vehicle classes based on Bina Marga vehicle types and then implemented in an application with object tracking methods such as the DeepSORT. The application will be able to accept video input and output the counting result as a file, which can be viewed once the object counting process is completed.

Following that, a separate evaluation will be conducted for object detection and object counting. The evaluation will be carried out for object detection using the Mean Average Precision (mAP) of each 1000th iteration while training. On the other hand, vehicle counting will be tested by comparing the manual counting result of vehicles and the results from the implemented application. The following are the steps done in this research.

Dataset Collection

It is required to gather data that will be utilised as training data to create a deep learning model that uses custom classes. The information will be presented as images of roads that will be used to analyse the input data. Images are manually captured or acquired from video recordings of vehicles cut into frames to obtain this information.

On local traffic survey websites, we obtain CCTV footage from various parts of Indonesia that can be used to create the dataset. A dataset in a jpg file is created from many frames of video streaming which is then prepared as a dataset for the YOLOv4 deep learning model.

Dataset Annotation

To prepare the dataset for YOLOv4, jpg images are manually searched for vehicles in the

frame and labeled for each vehicle found using the labelImg as shown in Figure 1. Annotation tool, which already has a setting for YOLOv4 labels and bounding boxes. An example of an annotated dataset image is shown in Figure 1.

The datasets that have been gathered will be separated into three categories: training, validation, and testing datasets. In general, deep learning uses a train-test split on its datasets, according to Dobbin and Simon [13], to reduce the issue of overfitting and to generalise the learning algorithm wellto cases that are not observable in future environments. The train-test split method ratio that is typically applied is 90-10. However, the dataset is split into an 80-10-10 ratio because it is divided into three parts (training, validation, and testing).

Model Training

Model training uses the Darknet framework and YOLOv4 pre-trained model, which can use a custom dataset and custom classes. The model is trained to properly detect every class using the annotated images in a dataset which categorises the training as supervised learning. Therefore, a set of hyperparameters is required to configure the training process, which helps the model create parameters fitting for the dataset. The hyperparameters used in this study are shown in Table 2.

Width and height are resolutions that will be used as the target image height after down-sampling. Max_batches is the maximum number of iterations of model training. The value of max_batches is calculated by the formula: classes x 2000 and a minimum of 6000. Filters are the numberof kernels used in each image convolution layer. The value of filters is calculated by the formula: (number of classes + 5) x 3.

And last but not least, steps are adjustments to the learning rate when the number of batches reaches each step's value. For example, the steps are 500 and 1000. Then the learning rate improvement is made when the batch reaches 500 and 1000. The number of steps is 80% and 90% of max_batches [14].

Figure 1. Annotation result of dataset image

Table 2. Hyperparameters for model training

Hyperparameter	Value
width	416
height	416
max_batches	24000
filters	51
steps	19200, 21600

Determining Confidence Threshold

The confidence threshold is the number that is used to decide whether or not detection is displayed, based on confidence which is a percentage of probability that the object is correctly classified. The confidence threshold is used as a parameter in a detection framework such as Darknet and can be determined by the True Positives and True Negatives ratio. Using Darknet's feature of displaying mAP after every 1000 iterations, the True Positives and True Negatives are also displayed along with the mAP.

Perform Vehicle Counting and Measure the Accuracy

To calculate how accurate the counting result of the application is, an estimation error or percentageerror is used to find how far the difference (by percentage) between the application's counting result is compared to the manual one. The application's counting result is used as the Approximate Value (A), and the manual counting result is used as the Exact Value (E), and the formula for estimation error is as in (1).

$$Error\ Percentage = \frac{|Approximate\ Value - Exact\ Value|}{|Exact\ Value|} \quad (1)$$

The percentage error could be used to subtract from 100 to depict the percentage of correctness to find the accuracy of the counting result. This method to count accuracy should use a video that the PUPR officially counts to produce the best results [15].

RESULTS AND DISCUSSIONTRAINING MODEL

Model training is carried out using the framework Darknet. The process will not be stopped if mAP continues rising [14]. The result can be seen in Figure 2. Whilst training, the mAP fluctuates around 50 - 70% but stagnates around 60%. Therefore, the training process stopped before it decreased further. As a result, the 4000th iteration is used for the detection model for having the highest mAP (66.17%).

Determining Confidence Threshold

The confidence threshold is a parameter for determining how confident deep learning is in detecting objects. The best threshold for detection using the available model can be determined by using one of the features that calculate mAP for each threshold in the Darknet and comparing True Positive with False Positive in the result. The result is shown in Table 3.

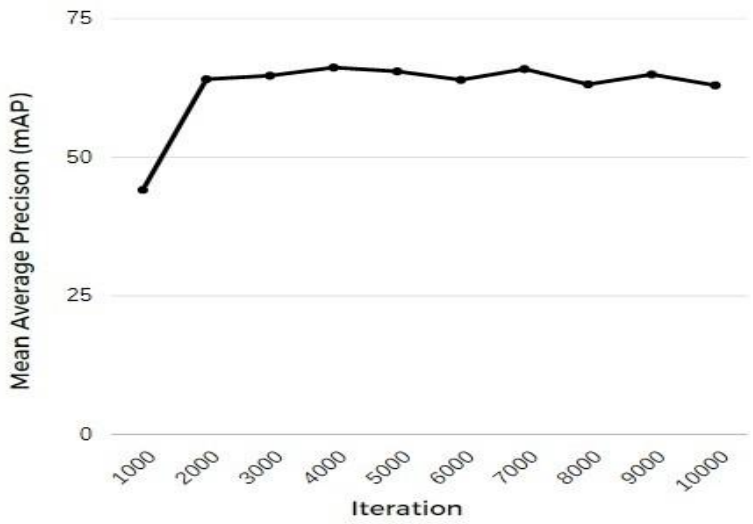

Figure 2. Training Result

Table 3. Confidence Threshold

Threshold	TP	FP
0.1	472	151
0.2	465	125
0.3	458	115
0.4	454	103
0.5	453	94
0.6	447	85
0.7	442	79
0.8	431	68
0.9	417	58
1.0	0	0

Table 4. Estimation Error Result

Time	AVG Accuracy per Interval	AVG Accuracy per Class	AVG Accuracy Total
Daytime	70.79	70.95	70.87
Nighttime	62.11	62.11	62.11
Combined	67.89	67.99	67.94

By dividing True Positive and False Positive, threshold 0.9 is chosen as the threshold for detection for having thehighest ratio of True Positive to False Positive (7.189).

Error Estimation Result for Counting Objects

After training the model and choosing the threshold, the next step is calculating actual results for detection and counting after converting the YOLO weights from the Darknet framework to be used in the TensorFlow framework. A one-hour-long video taken from Cipali Highway between 17:00 and 18:00 was used for testing; it features both daytime and nighttime. Table 4 displays accuracy results after calculating this model counting and classifying results with manual counting and vehicles from the video. It is also shown as accuracy with the formula: 100% subtracted by estimation error [15].

CONCLUSION

This study proposed a non-real-time vehicle detection system using model YOLOv4 and TensorFlow. YOLOv4 is used to localise and classify vehicles, and the Deep SORT algorithm is used for tracking objects. The dataset for training and testing is taken from local CCTV. Model training results indicate that the model reached mAP of 66.17% using the testing dataset, and counting results indicate that the model reached an average of 67.94%. Given the experiment's results, it is thought that this method can be applied to classify and count vehicles within a limited scope. However, in this study, there are a few limitations. Since this model is based on Indonesia's vehicle dataset, there is a significant chance that it will be inaccurate when counting and classifying vehicles on a highway in another nation.

Additionally, testing at night yields significantly lower accuracy due to the headlight glare that reflects on thecamera. Improvement in the range of the dataset, further study will improve the range and number of images for the dataset and eliminate problems such as weather, lighting, and object blur that may affect the detection accuracy. A faster network connection and more computing capabilities could also improve uploading time and detection speed respectively for the application to perform better.

ACKNOWLEDGMENT

We thank Mr. Yudi Widhiyasana, Mr. Didik Suwito Pribadi, and Mr. Djoko Cahyo Utomo

Lieharyani from Bandung State Polytechnic who provided insight and expertise that greatly assisted in verifying the research, although they may not agree with all of the interpretations/conclusions of this paper.

REFERENCES

[1] A. Mohammadnazar, R. Arvin, and A. J. Khattak," Classifying travelers' driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning," *Transportation Research Part C: Emerging Technologies*, vol. 122, ID: 102917, 2021, doi: 10.1016/j.trc.2020.102917

[2] Y. Wang et al., "Detection and Classification of Moving Vehicle From Video Using Multiple Spatio-Temporal Features," in *IEEE Access*, vol. 7, pp. 80287-80299, 2019, doi: 10.1109/ACCESS.2019.2923199.

[3] DPU, "Modul Rekayasa Lalu Lintas," *Jakarta Dep. Pekerja. Umum*, 2005.

[4] A. Ramya, V. Gupta Pola, A. Bhavya Vaishnavi, dan S. Suraj Karra, "Comparison of YOLOv3, YOLOv4 andYOLOv5 Performance for Detection of Blood Cells," *International Research Journal of Engineering and Technology(IRJET)*, vol. 8, no. 4, pp. 4225–4229, 2021.

[5] S. H. Jayady and H. Antong, "Theme Identification using Machine Learning Techniques", *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 1, no. 2, pp. 123-134, 2021, doi: 10.51662/jiae.v1i2.24

[6] A. Ashraf et al., "Detection of Road Cracks Using Convolutional Neural Networks and Threshold Segmentation," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 123-134, 2022, doi: 10.51662/jiae.v2i2.82

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, London, England: MIT Press, 2016.

[8] S. Dong, P. Wang and K. Abbas," A survey on deep learning and its applications," *Computer Science Review*, vol. 40, ID: 100379, 2021, doi: 10.1016/j.cosrev.2021.100379

[9] R. Jindal, A. Panwar, N. Sharma, and A. Rai, "Object tracking in a zone using DeepSORT, YOLOv4 and TensorFlow," in *2021 2nd International Conference for Emerging Technology (INCET)*, 2021.

[10] X. Zhang, X. Hao, S. Liu, J. Wang, J. Xu, and J. Hu, "Multi-target tracking of surveillance video with differential YOLO and DeepSort," in *Eleventh International Conference on Digital Image Processing (ICDIP 2019)*, 2019, doi: 10.1117/12.2540269

[11] X. Dong, S. Yan, and C. Duan, "A lightweight vehicles detection network model based on YOLOv5," *Engineering Applications of Artificial Intelligence*, vol. 113, ID: 104914, 2022, doi: 10.1016/j.engappai.2022.104914

[12] S. Kumar, Vishal, P. Sharma and N. Pal, "Object tracking and counting in a zone using YOLOv4, DeepSORT and TensorFlow," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021, pp. 1017-1022, doi: 10.1109/ICAIS50930.2021.9395971.

[13] K. K. Dobbin and R. M. Simon, "Optimally splitting cases for training and testing high dimensional classifiers," *BMC Medical Genomics*, vol. 4, no. 1, p. 31, 2011, doi: 10.1186/1755-8794-4-31

[14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv [cs.CV], 2020.

[15] C. J. Lin, S. Y. Jeng, dan H. W. Lioa, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO," *Mathematical Problems in Engineering*., vol. 2021, 2021, doi: 10.1155/2021/1577614.