International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

ROAD TRAFFIC ANALYSIS USING YOLO-V4 AND DEEP SORT

[1] B. SHOBHITHA, [2] G. NITHIN, [3] A.SAI, [4] K. ABHIVARAN,
[5] DR. K. VASANTH KUMAR, [6] MR. LALAM RAMU

Department of Computer Science and Engineering – Internet of ThingsMalla Reddy Engineering College, Hyderabad, Telangana saianumolu7@gmail.com

Abstract

The Ministry of Public Works and Public Housing (PUPR) undertook a traffic survey to ascertain the total number of cars and classify them using the Bina Marga vehicle categorization. The survey has so far been carried out manually. As a result, surveys need a significant amount of time and money to complete. Furthermore, as the survey scope expands, so will the demand for surveyors. As a result, a substitute that can carry out the survey procedure automatically and with acceptable accuracy is required. One possibility is to employ deep learning technology to detect and classify automobiles that can be used in apps. The program is developed as a web application that displays a summary of vehicle calculations and accepts video data from traffic cameras. The deep learning model utilized is YOLOv4, which has been trained to detect vehicle classifications such as Bina Marga vehicle types. The model was trained and tested on the Google Collab platform with Python and the Darknet framework. The YOLOv4 and Deep SORT approach with custom dataset achieved a reasonable accuracy of 67.94%, considering the limited 1000 photos used to train the model.

Keywords:
Deep Learning; Vehicle Detection and Classification; YOLOv4

INTRODUCTION

A traffic study was undertaken at the Ministry of Public Works and Public Housing (PUPR) by determining the total number of cars and categorizing them according to the Bina Marga vehicle categorization. The Ministry of PUPR incorporates traffic survey data into policy planning and report preparation. So far, surveyors have calculated and classified automobiles in traffic surveys manually, watching one vehicle lane on-site or using Closed-circuit Television (CCTV) camerarecordings. Given the time and money spent on each survey for computation and categorization, as well as the creation of a traffic survey report document, the manual traffic survey procedure is expensive.

Furthermore, there is a considerable risk of inaccuracies in the computation and categorization of cars produced by the Surveyors are required to identify and count various vehicle classifications atthe same time. It displays the vehicle classes, which are employed as detection classes in this investigation. Vehicle identification and tracking is a typical challenge with many applications.

Understanding traffic flow may help government and private establishments improve infrastructure for everyone's convenience. A road widening project's timetable

Analyzing traffic is crucial for projects like traffic lights and parking lot building. Traditionally, identification and tracking have been done manually. A person will count the number and kind of automobiles at a certain location. Sensors have been introduced, although they solely address counting issues. Sensors will not identify the vehicle type.

This paper proposes an alternate technique of traffic survey that might potentially lessen reliance on human surveyors by utilizing deep learning principles such as object identification, categorization, and tracking. A website-based application that conducts object detection, classification, and computation was presented to assist analyze the YOLOv4 and Deep SORT technique results using the dataset from CCTV mounted on the roadway.

YOLOv4 is employed despite the presence of YOLOv5 due to its superior performance attributesand shorter training time. Deep SORT was chosen because it has an average Multiple Object TrackingAccuracy (MOTA) of 68.7% and offers faster identification and tracking.

The web application will be able to take video files and upload them over an API for processing by the server at Google Collab. The server would then begin the detection, categorization, tracking, and counting ofthe video using YOLOv4. The Deep SORT technique requires that the YOLOv4 model be translated after training and utilized in the TensorFlow environment.

METHOD

The research is carried out by creating an application that counts the number of cars by category at predetermined intervals. To detect specific vehicle classifications, a custom-trained deep learning modelis required. As a result, the YOLOv4 model will be trained to recognize 12 vehicle classes based on Bina Marga vehicle types before being applied in an application that uses object tracking technologies likeDeep SORT. The program will be able to receive video input and output the counting results as a file that can be seen when the item counting procedure is done. Following that, a separate assessment will be performed for object recognition and counting. Object detection will be evaluated using the Mean Average Precision (MAP) of the 1000th iteration during training. Vehicle counting, on the other hand,will be tested by comparing the results of manual vehicle counting to those of the installed application.The following are the actions taken throughout this research.

Dataset Collection

It is necessary to collect data that will be used as training data to develop a deep learning model with specific classes. The information will be provided asphotographs of roadways, which will be utilized to analyze the input data. This information is obtained by manually capturing or acquiring images from video recordings of cars sliced into frames.
We obtained CCTV video from various locations of Indonesia that may be utilized to generate the datasetfrom local traffic survey websites. A dataset in jpg format is built from many frames of video streaming and then compiled as a dataset for the YOLOv4 deeplearning model.

Dataset Annotation

To prepare the dataset for YOLOv4, jpg photos are manually searched for automobiles in frame and tagged with Labeling, as illustrated in Figure 1. Theannotation tool already has a setting for YOLOv4 labels and bounding boxes. Figure 1 illustrates an annotated dataset image.
The datasets collected will be divided into three categories: training, validation, and testing datasets. According to Dobbin and Simon, deep learning utilizes a train-test split on its datasets in order to avoid overfitting and generalize the learning processto scenarios that are not visible in future contexts.
The train-test split technique ratio that is frequently used is 90-10. However, the dataset is separated into three sections (training, validation, and testing), resulting in an 80-10-10 ratio.

Model Training

The Darknet framework is used for model training, along with a pre-trained YOLOv4 model that can use a custom dataset and classes. The model is trained to correctly recognize each class using annotated photographs in a dataset, classifying the training as supervised learning. As a result, a collection of hyperparameters are necessary to design the training process, allowing the model to generate parameters that are appropriate for the dataset. The hyperparameters employed in this investigation are listed in Table 2.
Width and height are resolutions that will be used to determine the target picture height after down sampling. Max batches specifies the maximum number of model training iterations. Max batches is determined using the formula classes x 2000, with a minimum of 6000. Filters represent the number of kernels used in each picture convolution layer. The value of filters is computed using the following formula: (number of classes + 5) x 3.
Finally, steps change the learning rate when the number of batches approaches the step's value. For example, the steps are 500 and 1000. The learning rate improves as the batch approaches 500 or 1000. The number of steps is between 80% and 90% of the maximum number of batches.

Figure 1. Annotation result of dataset image

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

Table 2. Hyperparameters for model training

Determining Confidence Threshold

The confidence threshold is a value used to determine whether or not detection is shown, and it is calculated using confidence, which is a proportion of the chance that the item is properly categorized. The True Positives and True Negatives ratio may be used to calculate the confidence threshold, which is a parameter in detection frameworks like Darknet. Using Darknet's functionality of presenting mAP after 1000 iterations, the True Positives and True Negatives are presented with the mAP.

Perform Vehicle Counting and Measure the Accuracy

To determine how accurate the application's counting result is, an estimation error or percentage error is utilized to quantify the difference (by percentage) between the application's counting result and the manual one. The application's counting result is utilized as the Approximate Value (A), while the manual counting result is used as the Exact Value (E), with the estimate error formula given in (1).

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

The percentage error might be subtracted from 100 to represent the percentage of correctness in order to determine the accuracy of the counting result. To achieve the greatest results, this technique of counting accuracy should be used with a movie that the PUPR officially counts [15].

Determining Confidence Threshold

Model training is conducted out utilizing the Darknet framework. If mAP continues to rise, the process will continue [14]. Figure 2 shows the result. During training, the mAP varies between 50 and 70% but stabilizes at 60%. As a result, the training process came to an end before it could progress any further. Consequently, the 4000th iteration is used for the detection model with the greatest mAP (66.17%).

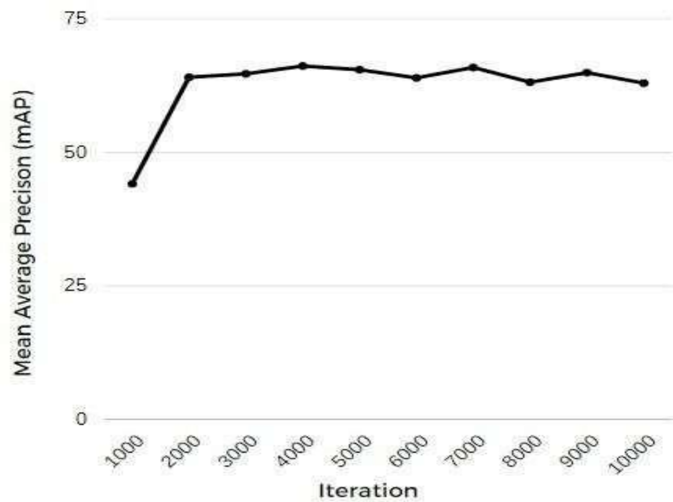

figure 2. Training Result

Threshol d	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 3. Confidence Threshold

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

Table 4. Estimation Error Result

Error Estimation Result for Counting Objects

After training the model and selecting the threshold, the following step is to calculate the actual detection and counting results after transferring the YOLO weights from the Darknet framework to TensorFlow. Testing was conducted using a one-hour video shot from Cipali Highway between 17:00 and 18:00, which includes both daylight and nighttime footage. Table 4 shows the accuracy results obtained by computing this model counting and categorizing results with manual counting and cars from the video. It is also represented as accuracy using the formula: 100% subtracted by estimation error [15].

REFE R

CONLUSION

This research introduces a non-real-time system for vehicle detection employing YOLOv4 model and TensorFlow. YOLOv4 locates and categorizes vehicles, while object tracking is handled by the Deep SORT algorithm. The dataset, sourced from local CCTV, is utilized for both training and testing purposes. The model achieves a mean Average Precision (mAP) of 66.17% on the testing dataset, with an average counting accuracy of 67.94%. These findings suggest the potential applicability of this approach for vehicle classification and counting within specific contexts. However, the study identifies several limitations. Since the model is trained on an Indonesian vehicle dataset, its accuracy may suffer when applied to highways in other countries. Moreover, nighttime testing results in reduced accuracy due to glare from headlights affecting camera visibility. To address these limitations, futu work will focus on expanding the dataset toencompass a broader range of scenarios and improving detection accuracy by mitigating issues such as weather, lighting variations, and object blurring. Additionally, enhancements in network connectivity and computational capabilities are suggested to optimize upload and detection speeds,respectively, thereby enhancing overall system performance.

REFERNCES

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

[1] 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.

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

[3] 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.

[4] 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

[5] A. Ashraf et al., "Detection of Road Cracks Using Convolutional Neural Networks and Threshold Segmentation,"*Journal of Integrated and Advanced*

[6] *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.

[17] Dr. Rugada Vaikunta Rao, Mrs. A.Laxmi Prasanna , Mr. Gugloth Ganesh , Mrs.Vadla Anuja ,Mr.Konatala Lokesh , Dr.K.Vasanth Kumar. (2023). Securing Healthcare: A Fusion of AI and Blockchain for Medical Data Protection. Journal of Advanced Zoology, 44(S2), 1396–1405. https://doi.org/10.53555/jaz.v44iS2.975