ASTONJADRO

pISSN 2302-4240
eISSN 2655-2086

Volume 13, Issue 1, February 2024, pp.115-124
DOI: http://dx.doi.org/10.32832/astonjadro.v13i1 http://ejournal.uika-bogor.ac.id/index.php/ASTONJADRO

Traffic Counting using YOLO Version-8 (A case study of Jakarta-Cikampek Toll Road)

Darmadi Darmadi[1,2], Pratikso Pratikso[3], Mudiyono Rachmat[3]

[1]Civil Engineering Doctoral Program Sultan Agung Islamic University, Semarang, INDONESIA
[2]Department of Civil Engineering, University of Jayabaya, Jakarta, INDONESIA
[3]Civil Engineering Department, Sultan Agung University, Semarang, INDONESIA

E-mail: darmadi@std.unissula.ac.id

| Submitted: June 18, 2023 | Revised: June 10, 2023 | Accepted: December 11, 2023 |

| Published: January 12, 2024 |

ABSTRACT

You Only Look Once (YOLO) version 8 is the latest version of YOLO. YOLO is a common object detection model that offers faster and more accurate results. YOLO applications provide numerous benefits in the fields of health care, traffic control, vehicle safety, energy, agriculture, and industry. The purpose of this article is to use advancements in information technology to automate the process of manually recording traffic counts on the highway. The method utilized in this study is to record a video of traffic movements with a smartphone camera and save it in MP4 format. Calculations are performed at the office after receiving recorded video and utilizing a program written by the author that makes use of Python, Ppencv, Pytorch, and YOLO version 8 software. When passing through a counter box, the traffic volume is counted and saved in Excel format (.xls). The video records footage near the Halim area of the Jakarta-Cikampek toll road. With a measurement accuracy of 99.63% for cars, 96.66% for buses, and 98.55% for trucks, the accuracy attained using YOLO version 8 is fairly satisfactory for detecting vehicle volume and categorization.

Keywords: YOLO version 8; traffic volume; Opencv; smartphone; Python and Pytorch.

INTRODUCTION

The advancement of information technology(Dillmann & Huck, 1985; Ismail et al., 2021) (particularly the emergence of artificial intelligence) stimulates the application of work assignments in practically all disciplines and even substitutes human labor. This advancement also relates to transportation, particularly the use of traffic computations with the development of object-detecting sensors, namely You Only Look Once (YOLO)(Gündüz & Işık, 2023; Redmon et al., 2016; Sauqi, 2022). As shown in **Figure 1 and Table 1.**, numerous object detection methods have been developed, including CNN, RCNN, and YOLO(Kim et al., 2020).

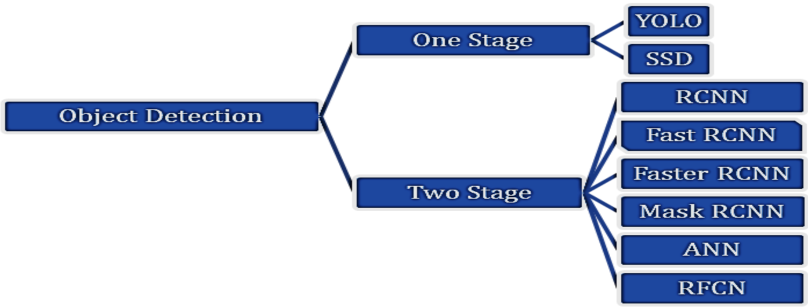

Figure 1. Type of Object Detection Method

Table 1. Comparation Parameter Object Deetection Method

no	Method	Backbone	Input Size	AP	AP.5	AP.75	APs	APm	APL
1	YOLO	Dak-Net	513	31,20	50,40	33,30	10,20	34,50	49,80
2	SSD	ResNet	513	33,20	53,30	35,20	13,00	35,40	51,10

3	DSSD	ResNet	512	36,40	57,50	39,50	16,60	39,90	51,40
4	RCNN	ResNet-101	512	34,90	55,70	37,40	15,60	38,70	50,90
5	Fast RCNN	ResNet	512	34,70	55,50	36,70	13,50	38,10	52,00
6	Faster RCNN	ResNet	512	36,80	57,70	39,20	16,20	39,80	52,10

AP = Average Precission. Source: (Lohia, 2021; Lu et al., 2020; Shreyas Dixit et al., 2019; Soviany & Ionescu, 2018)

YOLO makes predictions about the location and existence of objects. It uses a single pass of the input image and processes an image by utilizing a fully convolutional neural network (CNN), which boosts computing speed. YOLO suggests employing an end-to-end neural network to forecast bounding boxes and class probabilities simultaneously. By taking a fundamentally different approach to detecting an object, YOLO outperforms existing real-time object detection algorithms by a wide margin.

Single-shot object detection (SSD) is similar to YOLO in that it uses a single pass of the input image to make predictions about the location and existence of objects. In doing so, processing a full image in a single pass is more efficient. In fact, single-shot object detection is less accurate in general than other approaches.

Two-shot object detection makes predictions about the presence and placement of objects using two passes of the input image. The *first* pass generates a set of proposals or prospective object locations, while the *second* pass refines these recommendations and makes final predictions. This method is more accurate than single-shot object detection. However, it is also more computationally expensive and time-consuming. The application's specific goals and constraints also have an impact on the choice between single-shot and two-shot object detection.

Single-shot object detection is generally better suited for real-time applications, but two-shot object detection is better suited for situations where precision is more critical. YOLO executes all of its predictions with the help of a single fully connected layer, instead of detecting prospective regions of interest and then doing recognition on those regions separately.

The manual computation method was typically used to calculate traffic volume, with individuals counting directly on the roadway with counter equipment (**Figure 2**). Another method of determining traffic volume is to install measuring pipe tubes alongside the road. The pipe measures the change in air pressure brought about by vehicle weight (**Figure 3**). Installing loops that use variations in electromagnetic waves as a counter to the number of vehicles and vehicle weight is a better method today (**Figure 4**), and in the future, it will undoubtedly develop by utilizing technological advancements in detecting vehicles on recorded video images of vehicles and performing calculations in the office by YOLO (**Figure 5**).

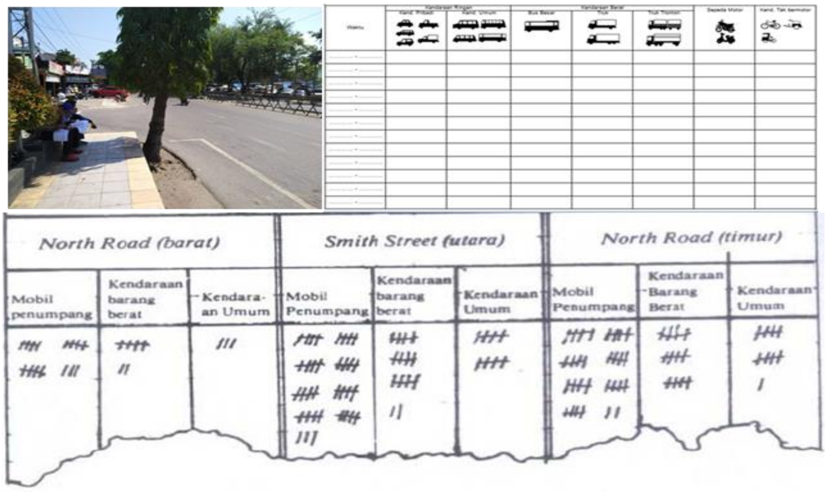

Figure 2. Conventional traffic count

Volume 13, Issue 1, February 2024, pp.115-124
DOI: http://dx.doi.org/10.32832/astonjadro.v13i1 http://ejournal.uika-bogor.ac.id/index.php/ASTONJADRO

Figure 3. Air pressure pipe across the road

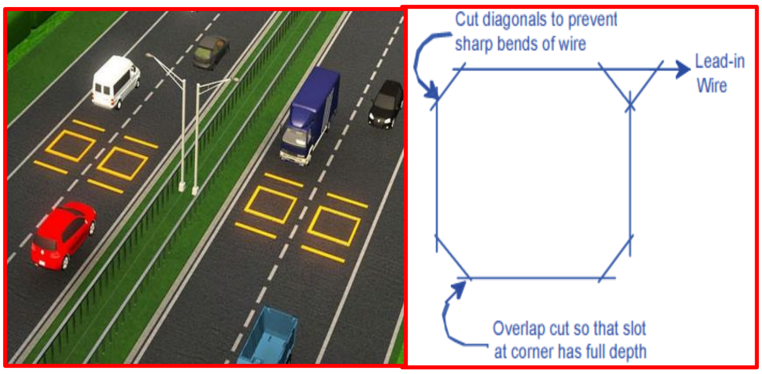

Figure 4. Coil loop method traffic count

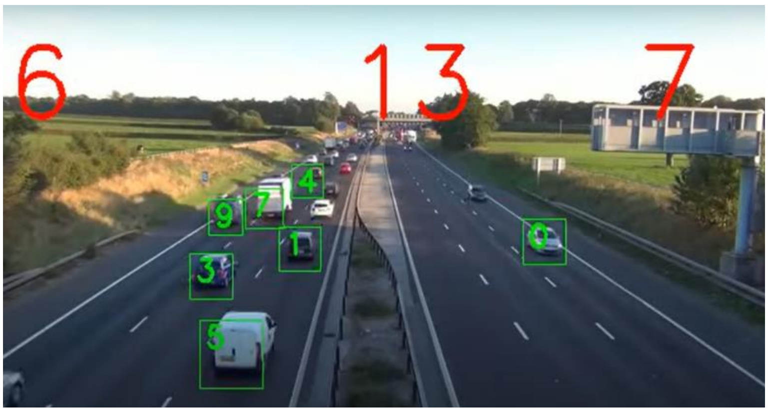

Figure 5. Traffic count using Computer Vission

Artificial intelligence and machine learning advancements(ChangjiangZheng & JianJohnLu, 2009; Dillmann & Huck, 1985; HonggyuLee, n.d.; Ismail et al., 2021; Lee & Chen, 2023), including deep learning techniques such as PyTorch, Deepsort, Tensorflow, YOLO (You Only Look One), and others, will make digital traffic volume estimations easier and more efficient. YOLO is a computer vision tool for recognizing things that is becoming increasingly useful in a variety of applications. YOLO is a free (open source) model for quickly and precisely recognizing objects. When compared to YOLO v5 and before, the detection ability of the model in the latest YOLO version has increased in terms of detection accuracy and speed, as well as image quality detection capabilities. The YOLO program is currently widely utilized in the robotics sector, the development of surveillance tools, the development of self-driving automobiles, and a variety of other fields.

YOLO detects it using the convolutional neural network (CNN) approach (Stanley, 2021), which is as close to human thinking as artificial intelligence (AI) (Lee & Chen, 2023). As a result, YOLO employs data used for introduction or training (trained), and the subsequent process employs additional data as data for the validation process, resulting in an object classification that now detects 80 objects(Lin et al., 2014; Wen et al., 2023). YOLO version 8 is a popular open-source vision-level Artificial Iintelligent application developed by Ultralytics. Because the technique is simple, it is possible that it will become a visionary AI method that is always utilized in the domains of college learning, autonomous driving, automotive industry development, and others in the future. Several new versions of the same model have been proposed since the initial release of YOLO in 2015. **Figure 6** is a timeline showcasing YOLO's development until now.

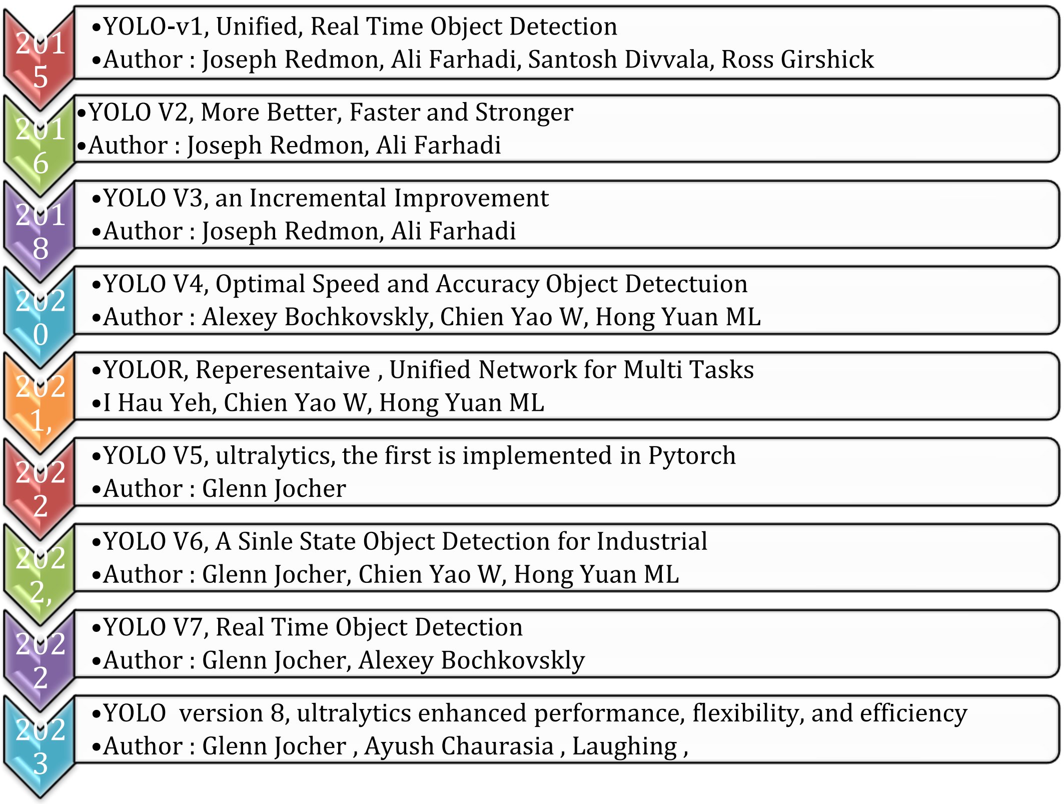

Figure 6. Evolution or Transformation of YOLO

The comparation of speed parameter between YOLO v5, YOLO v6, YOLO v7 and YOLO version 8 is presented in **Figure 7**, while comparison of improvement the accuracy detection vehicle is presented in **Figure 8**.

ASTONJADRO pISSN 2302-4240
eISSN 2655-2086
Volume 13, Issue 1, February 2024, pp.115-124
DOI: http://dx.doi.org/10.32832/astonjadro.v13i1 http://ejournal.uika-bogor.ac.id/index.php/ASTONJADRO

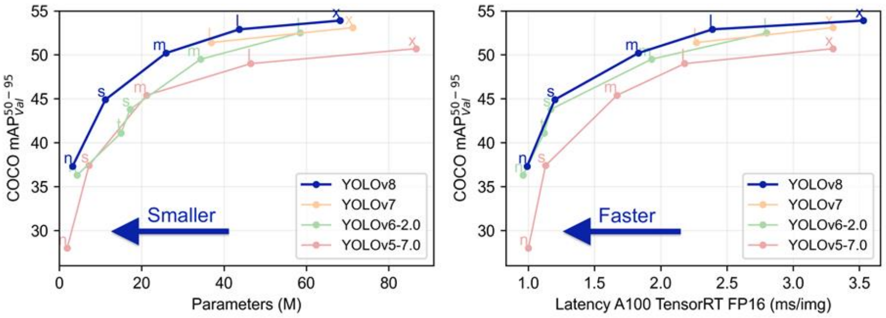

Figure 7. Comparation of YOLO

In 2015, YOLO created an object detection system that yielded a total of 80 (eighty) object classifications from identification and validation utilizing Microsoft Common Objects in Context (COCO) data. Yolo (Redmon & Farhadi, 2018) is a complex real-time object recognition method developed in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi and previously trained on the COCO dataset. It processes the entire image with a single neural network. The image is divided into boxes, each with its own method for estimating probability and bounding boxes. The application of YOLO version 8 will be investigated in this study to detect items, particularly automobiles, which are only detected in five kinds of vehicles, namely motorbikes, cars, buses, and trucks.

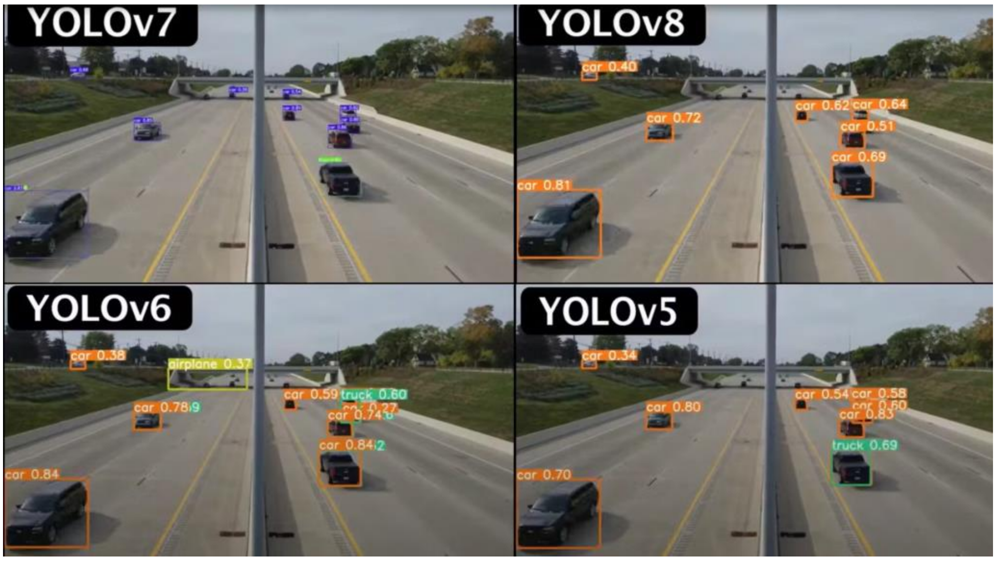

Figure 8. The accuracy different of YOLO detected the objects Source: https://www.youtube.com/watch?v=QOC6vgnWnYo

RESEARCH METHODS
Materials

When compared to previous studies, the focus of this research differs in identifying the volume segregated in one lane and saving the computation findings in .xls or .csv format. This research was conducted in Jakarta – Cikampek Toll road, specifically at Jatiwaringin, Bekasi City, West Java, in April 2023. The location of survey can be seen in **Figure 9.** The placement of the location of the camera and the equipment used can be seen in **Figure 10.** the stages of the research stages are presented in the **Figure 11.**

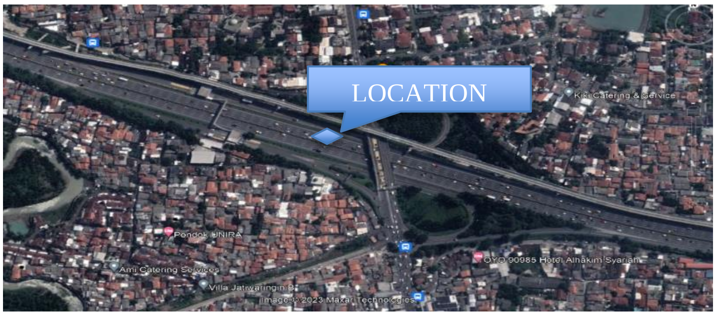

Figure 9. Location of Study

Figure 10. Position of Camera

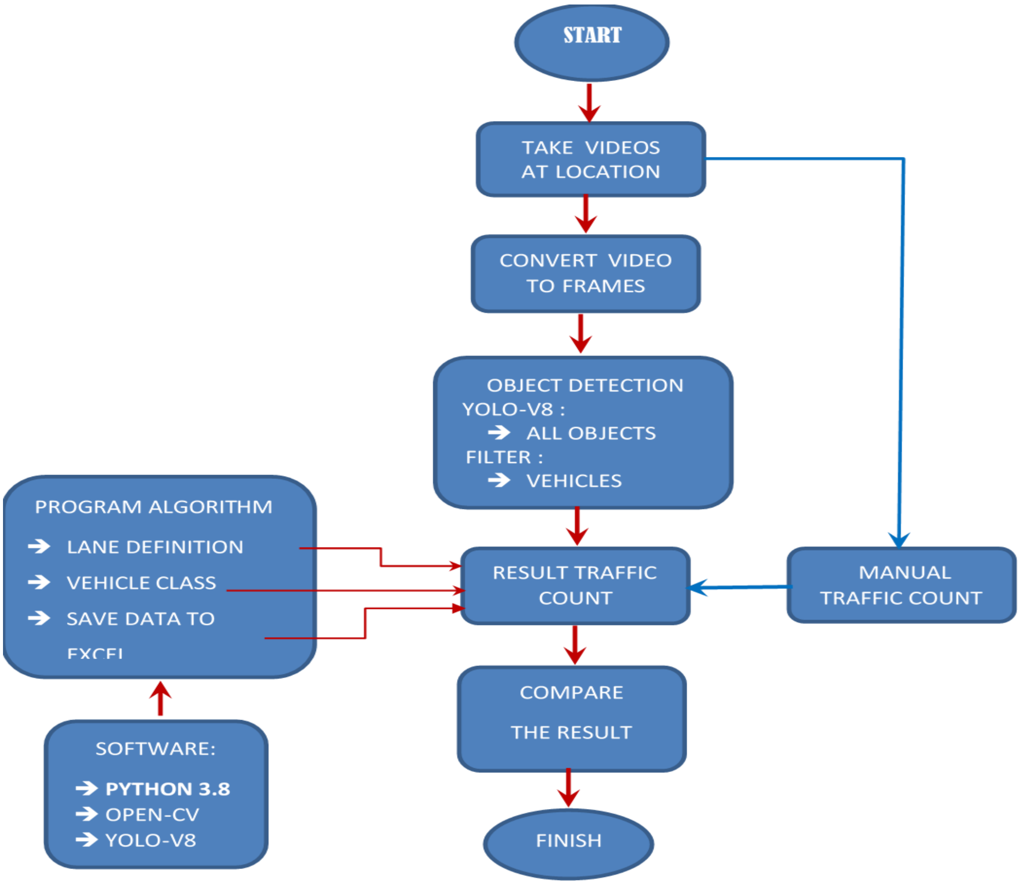

Figure 11. Research method

ASTONJADRO
pISSN 2302-4240
eISSN 2655-2086
Volume 13, Issue 1, February 2024, pp.115-124
DOI: http://dx.doi.org/10.32832/astonjadro.v13i1 http://ejournal.uika-bogor.ac.id/index.php/ASTONJADRO

Methods

This study explores how YOLO version 8 is utilized in transportation studies, specifically how to compute highway traffic volume. Software algorithms like Python(Asha & Narasimhadhan, 2018; Paszke et al., 2019; Redmon & Farhadi, 2018; Stanley, 2021), OpenCV , NumPy , Pytorch, and OpenXL(ChangjiangZheng & JianJohnLu, 2009; Gazoni, 2015) are used to do calculations. The results of the calculations will be saved in Excel format (.xls) using the function OpenXL.

The study began with the collection of primary data, specifically video data, in the field using a smartphone with a resolution of 1200 pixels and a memory card with a capacity of 128 GB for storing recordings. Video samples were obtained by filming them on the scene and then processing them with a computer at the office.

Data Analysis

Data analysis was carried out by software application. OpenCV, Python, OpenXL, and YOLO version 8 all helped create software applications. **Figure 12** depicts a description of software application algorithms. This study used the YOLO version 8 software application to analyze data from videos recorded from the Jakarta-Cikampek toll road in order to quantify the traffic volume on the roadway. Software algorithms such as Python, YOLO, OpenCV, NumPy, and OpenXL are used to do calculations of Traffic volume. The calculation results will be saved in an excel file using the OpenXL function. The goal of this study was to compare the results to manual data traffic counts and volume traffic counts by YOLO version 8 in a different lane.

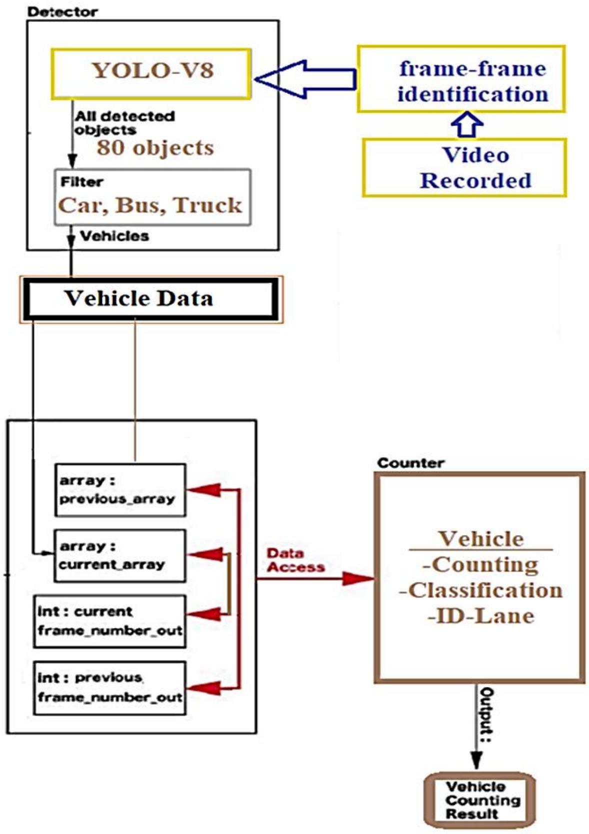

Figure 12 Algoritm Software Application

RESULT AND DISCUSSION

The results of a survey of video capture of traffic conditions on the Jakarta-Cikampek Toll Road, Jatiwaringin, Bekasi City, West Java [30] acquired video data with a duration of 8 (eight) hours. The video data obtained is traffic data collected during the morning rush hour for two hours, from 07.00

to 16.00 (Western Indonesia Time). The initial step before continuing to count the number of vehicles passing through a road lane is to mark a box at the beginning of the frame, which will be used to mark the lane through which the vehicle is passing, as shown in **Figure 13.**

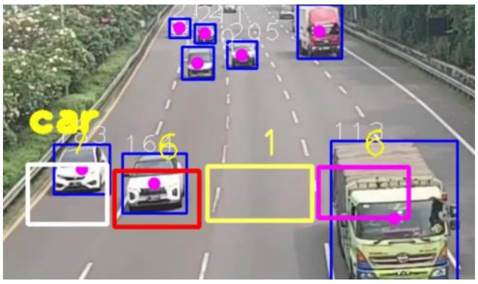

Figure 13. Traffic counts by using open access software/applications

The results of calculating traffic volume were provided in Excel format and divided traffic volume in each lane and also in their classification, as it can be seen in **Table 2.**

Table 2. Traffic count result (Day: Monday, April 10,2023)

Software Counting Results			
Time	**Car**	**Bus**	**Truck**
7:00 - 8:00	2767	204	1985
8:00 - 9:00	3114	238	2020
9:00 - 10:00	2453	92	229
10:00 - 11:00	2833	84	216
11:00 - 12:00	2447	114	1758
12:00 - 13:00	2803	153	1794
13:00 - 14:00	3250	259	607

Manual calculations are also performed to compare the level of accuracy acquired from software versus manual calculations. **Table 3** shows the results of determining the volume of vehicle traffic on the Jakarta-Cikampek toll road at Jatiwaringin between 7:00 a.m. and 14:00 p.m.

Table 3. Comparation of Manual Traffic count and Software YOLO version 8

Day of Monday										
Manual Counting Result				Software Counting Results				**% Difference**		
Time	**Car**	**Bus**	**Truck**	**Time**	**Car**	**Bus**	**Truck**	**Car**	**Bus**	**Truck**
7:00 - 8:00	2780	206	1999	7:00 - 8:00	2767	204	1985	99,53%	99,03%	99,30%
8:00 - 9:00	3130	239	2033	8:00 - 9:00	3114	238	2020	99,49%	99,58%	99,36%
9:00 - 10:00	2477	93	228	9:00 - 10:00	2453	92	229	99,03%	98,92%	100,44%
10:00 - 11:00	2822	87	230	10:00 - 11:00	2833	84	216	100,39%	96,55%	93,91%
11:00 - 12:00	2457	113	1771	11:00 - 12:00	2447	114	1758	99,59%	100,88%	99,27%
12:00 - 13:00	2817	152	1803	12:00 - 13:00	2803	153	1794	99,50%	100,66%	99,50%
13:00 - 14:00	3255	254	619	13:00 - 14:00	3250	259	607	99,85%	101,97%	98,06%
						AVERAGE =		**99,63%**	**99,66%**	**98,55%**

Volume 13, Issue 1, February 2024, pp.115-124
DOI: http://dx.doi.org/10.32832/astonjadro.v13i1 http://ejournal.uika-bogor.ac.id/index.php/ASTONJADRO

CONCLUSION

With measurement accuracy of 98.74% for cars, 92.69% for buses, and 96.85% for trucks, the accuracy attained using YOLO version 8 for detecting vehicle volume and categorization is fairly satisfactory. In future, this paper suggests that vehicles be classified according to the Bina Marga 7 (seven) standard traffic classifications: motorcycle, automobile, small bus, large bus, small truck, medium truck, and large truck. According to the error less than 5%, it is sugested using YOLO version 8 for calculating traffic volume.

ACKNOWLEDGEMENT

The author is thanks to Prof. Ir. Pratikso, PhD and Ir. Rachmat Mudiyono, MT, PhD for helping me with thoughtful insight and feedback.

REFERENCES

Asha, C. S., & Narasimhadhan, A. V. (2018). Vehicle Counting for Traffic Management System using YOLO and Correlation Filter. *2018 IEEE International Conference on Electronics, Computing and Communication Technologies, CONECCT 2018*, *March*. https://doi.org/10.1109/CONECCT.2018.8482380

ChangjiangZheng, P. D., & JianJohnLu, P. D. ,P. E. ,Professo. (2009). *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE.

Dillmann, R., & Huck, M. (1985). Intelligent Simulation of Robot Application. *IFAC Proceedings Volumes*, *18*(16), 547–555. https://doi.org/10.1016/s1474-6670(17)60021-1

Gazoni, E. (2015). *openpyxl Documentation*. 1–273.

Gündüz, M. Ş., & Işık, G. (2023). A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models. *Journal of Real-Time Image Processing*, *20*(1). https://doi.org/10.1007/s11554-023-01276-w

HonggyuLee, D. Y. T. K. H. K.-H. H. (n.d.). *Predictive Intelligent Driver Model using Deep Learning-based*.

Ismail, E., Ayoub, B., Azeddine, K., & Hassan, O. (2021). Machine learning in the service of a clean city. *Procedia Computer Science*, *198*, 530–535. https://doi.org/10.1016/j.procs.2021.12.281

Kim, J. A., Sung, J. Y., & Park, S. H. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. *2020 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2020*, 8–11. https://doi.org/10.1109/ICCE-Asia49877.2020.9277040

Lee, M. F. R., & Chen, Y. C. (2023). Artificial Intelligence Based Object Detection and Tracking for a Small Underwater Robot. *Processes*, *11*(2). https://doi.org/10.3390/pr11020312

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 LNCS*(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Lohia, A. (2021). *DigitalCommons @ University of Nebraska - Lincoln Bibliometric Analysis of One-stage and Two-stage Object Detection Bibliometric Analysis of One-stage and Two-stage Object Detection*. *February*.

Lu, X., Li, Q., Li, B., & Yan, J. (2020). MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12359 LNCS*, 541–557. https://doi.org/10.1007/978-3-030-58568-6_32

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*(NeurIPS).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). YOLO v.3. *Tech Report*, 1–6. https://pjreddie.com/media/files/papers/YOLOv3.pdf

Sauqi, M. (2022). Deteksi Kendaraan Menggunakan Algoritma You Only Look Once (YOLO) V3. *Universitas Islam Indonesia*, 5–8.

Shreyas Dixit, K. G., Chadaga, M. G., Savalgimath, S. S., Ragavendra Rakshith, G., & Naveen Kumar, M. R. (2019). Evaluation and evolution of object detection techniques YOLO and R-CNN. *International Journal of Recent Technology and Engineering*, *8*(2 Special issue 3), 824–829. https://doi.org/10.35940/ijrte.B1154.0782S319

Soviany, P., & Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *Proceedings - 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018*, 209–214. https://doi.org/10.1109/SYNASC.2018.00041

Stanley, A. (2021). *6 with YOLO Algorithm and Road HSV Color Model- based Segmentation System Development*. *C*.

Wen, G., Li, S., Liu, F., Luo, X., Er, M. J., Mahmud, M., & Wu, T. (2023). YOLOv5s-CA: A Modified YOLOv5s Network with Coordinate Attention for Underwater Target Detection. *Sensors (Basel, Switzerland)*, *23*(7), 1–14. https://doi.org/10.3390/s23073367