IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

Detection of Violations by Two-Wheeled Vehicles on Pedestrian Roads Using YOLOv5

Hendro Darmono[1], Rachmad Saptono[2], Firman Firdaus Alamanta[3]

Digital Telecommunication Network, Department of Electrical Engineering, State Polytechnic of Malang,

Malang City, 65141, Indonesia[1,2,3]

Abstract: Violation of motorcyclists in special road areas such as pedestrian paths, bus lanes, and toll roads still frequently occurs, posing the risk of accidents and disrupting other road users. Factors contributing to these violations include avoiding traffic jams to reach destinations more quickly. Therefore, a motorcycle detection system has been developed to facilitate the handling of violations by providing information to relevant security personnel. The detection system utilizes the YOLOv5 method, trained with 492 images with varying parameters such as epoch value (200) and batch size (24). Using the YOLOv5 model, it achieves a Mean Average Precision (MAP) value of 89.8%, indicating good detection quality. The detection system can be implemented in real-time using a webcam. Dummy violation data is processed using a Raspberry Pi 4B microcontroller. Testing the detection system in light intensity ranging from 6171 Lux to 140516 Lux demonstrates its quick response capability in sending information via the Telegram application, taking around 171 milliseconds for each data packet. The system effectively detects objects and promptly provides information to relevant security personnel, enhancing performance in addressing violations within the monitored area.

Keywords: Object Detection, Raspberry Pi, Telegram Bot, Traffic Violations, YOLOv5

I. INTRODUCTION

The popularity of motorcycle usage in developing countries, such as Indonesia, is increasing due to its everyday convenience in flexibility, accessibility, affordability, and good maneuverability [1]. The growing number of motorcycle riders leads to violations, including those occurring in non-motorized road areas. Non-motorized road areas are roads where motorcycle riders are not allowed, such as pedestrian pathways where violations are often found in major cities. Additionally, violations occur on toll roads and bus lanes, used by larger passenger and cargo vehicles that have larger dimensions and relatively faster speeds than motorcycles. These violations are committed to avoiding traffic jams and reducing travel time but have consequences, resulting in accident risks and disrupting the smooth flow of traffic for other road users [2].

The increase in the mobilization of motorcycle users must be balanced with the development of traffic management. The development aims to understand the behavior of motorcycle riders in various situations, reducing accident risks and enhancing the safety of other road users [3]. Traffic management development can be achieved by improving information delivery systems that can transmit violation information to traffic management authorities. Efforts to develop information services to assist authorities in monitoring a region can implement an automatic violation detection system capable of sending information.

Previous research developed an information system for detecting violations of car and motorcycle drivers used on Transjakarta bus lanes, utilizing objects such as cars, buses, and motorcycles [4]. This system used dummy data to identify vehicle license plates with the help of the Tesseract OCR library. The devices used in this system included the Raspberry Pi 3 Model B as a microcontroller, an ultrasonic sensor HCSR04, and a webcam. The system was integrated with a website serving as a violation information center. When the devices were turned on, the ultrasonic sensor HC-SR04 would detect objects 10 cm in front of it, affecting the webcam's ability to capture images of the vehicle plates. The captured images were processed using the Tesseract OCR library with Raspberry Pi through thresholding to make them readable. However, the system had weaknesses as it only used the HC-SR04 sensor to detect moving objects, and the webcam was not optimal in reading license plate models if the object was more than 10 cm away and moving quickly. Based on the shortcomings identified in the previous research system for delivering violation information, this study aims to implement a violation detection system with motorcycle and person classes, applying a real-time webcam combined with object detection algorithms using dummy violation data. This system utilizes the YOLOv5 (You Only Look Once Version 5) method, capable of detecting vehicles through captured video frames and sending information via the Telegram application interface, including violation details and object images. The results of this research are expected to contribute to the development of technology focused on improving traffic safety.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

II. RESEARCH METHODOLOGY

The system works as illustrated by the block diagram in this research, depicted in Figure 1. Figure 1 provides an explanation of the system model's block diagram that will be created. A detailed explanation of each device used is as follows: (1) The web camera module functions as a capture sensor image to detect objects, and its data will be processed by the Raspberry Pi 4b. (2) The Raspberry Pi 4b, a mini PC, serves as a data processor between the input and output systems. (3) The Access Point serves as a connector for the Raspberry Pi 4b to the internet network, connecting to the Telegram application using an API token. (4) Telegram functions as a medium for receiving processed data information from the Raspberry Pi 4, including descriptions and images of violations sent to officers.

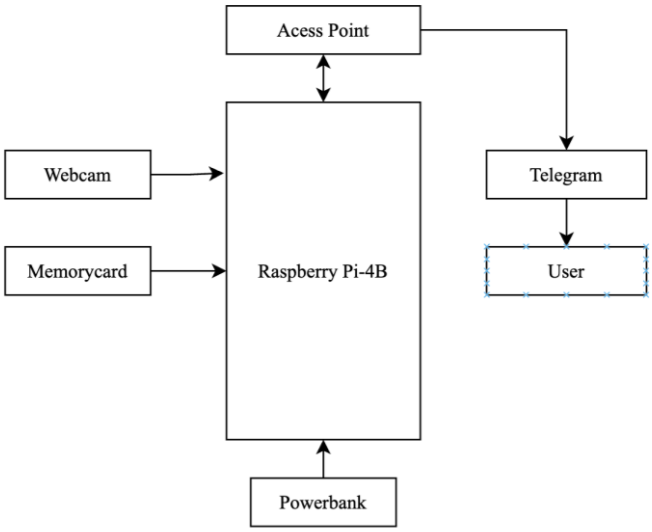

Fig. 1 System Block Diagram

A. System Flowchart

Fig. 2 represents the flowchart of the system, focusing on motorcycle objects and their riders. This dataset serves as input data that will be trained to obtain a model for detection. The training process of the dataset begins by assigning labels to the images, where labeling is used to distinguish objects in each class. In this stage, the roboflow website platform is used, providing convenience in labeling objects such as motorcycles labeled as the Motor class and riders labeled as the Person class. Once all datasets have been labeled, roboflow trains the data to annotate the objects [5]. These annotations include bounding boxes, class types, and accuracy detection scores. The output generated by roboflow is a snippet used as input data in Google Colab for YOLO (You Only Look Once) training. During the training process in Google Colab, the data is directly trained using the YOLOv5 library to improve accuracy by experimenting with variations in batch and epoch values [6]. The best accuracy value obtained will be exported as a file, best.pt. This file will be tested with input video to generate object detections. Detecting an object using the YOLO (You Only Look Once) algorithm has several processes [7]. Fig. 3 is a flow diagram of detecting an object with YOLO and sending information to Telegram. The process of object detection begins using the YOLO algorithm, where the input is video from the camera [8].

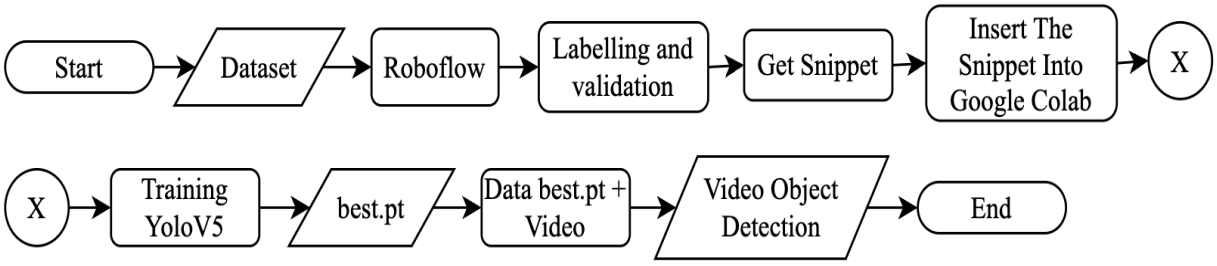

Fig. 2 Flowchart Data Preprocessing And Training Process

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

Next, the system will match the images that have been detected from the camera with YOLO training data, which results in object detection according to the class contained in the video [9]. Then, if the system results detect the class of motorbike and person, this data will take an image which is used to produce violation information. This information contains a description of the violation and an image that will be sent via the Telegram application interface.

B. Dataset Retrieval

A dataset is a collection of object data in the form of images of the objects to be detected. In this research, motorised vehicles and their riders are the objects to be detected. Fig.4 represents a dataset obtained from a video produced by a webcam camera, which is extracted into several frames to create a sequence of images. Converting the video into images is performed using VLC Player software.

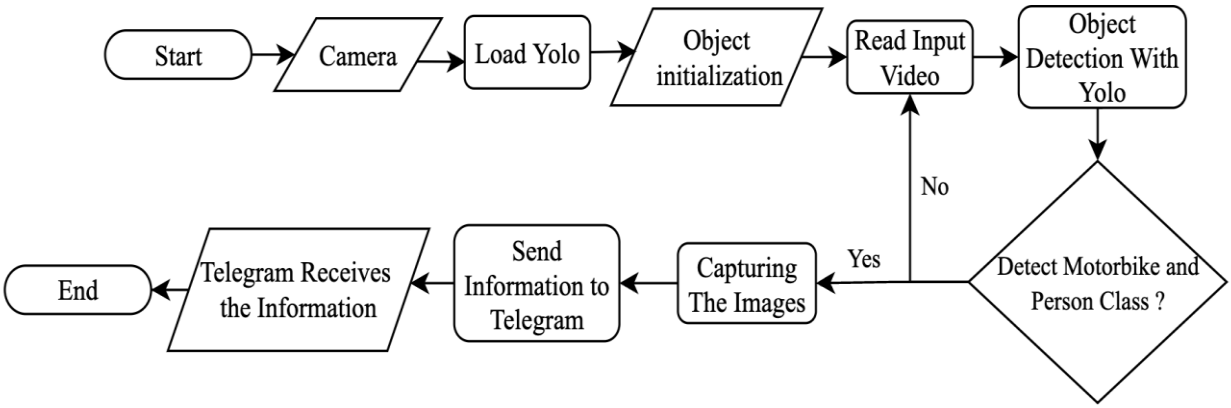

Fig. 3 Object Detection System Flowchart

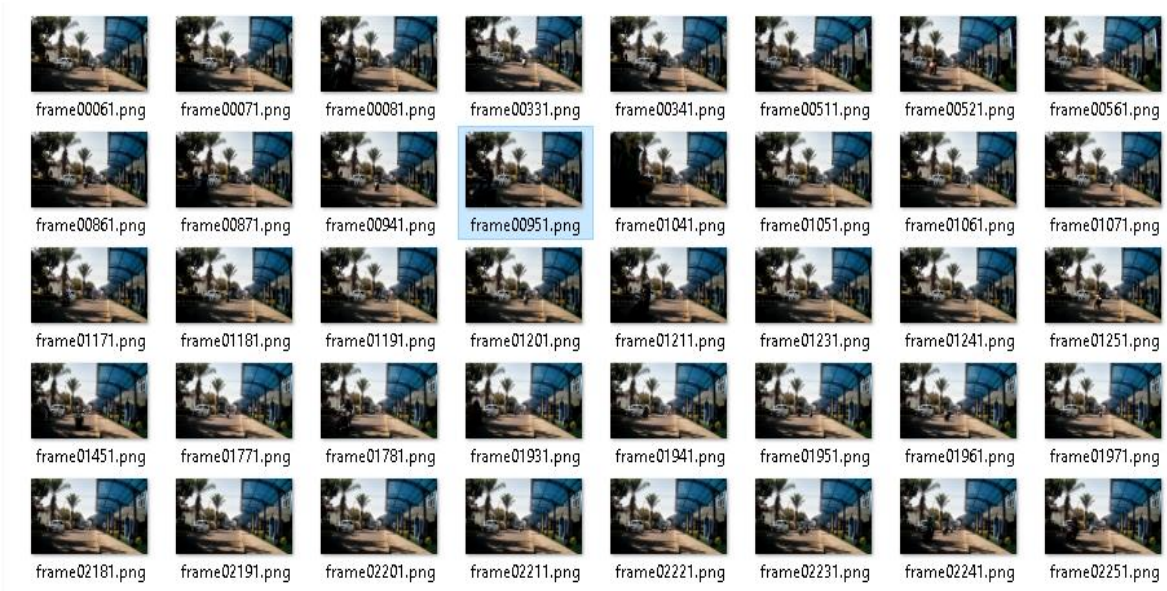

Fig. 4 Dataset retrieval

C. Label Creation

Label creation involves marking information on images to train object detection models. The process includes defining bounding boxes around objects and assigning class labels to each visible object. The results of labeling provide the system with the ability to recognize and understand objects in images. Fig. 5 illustrates the process of labeling objects using the Roboflow website platform.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

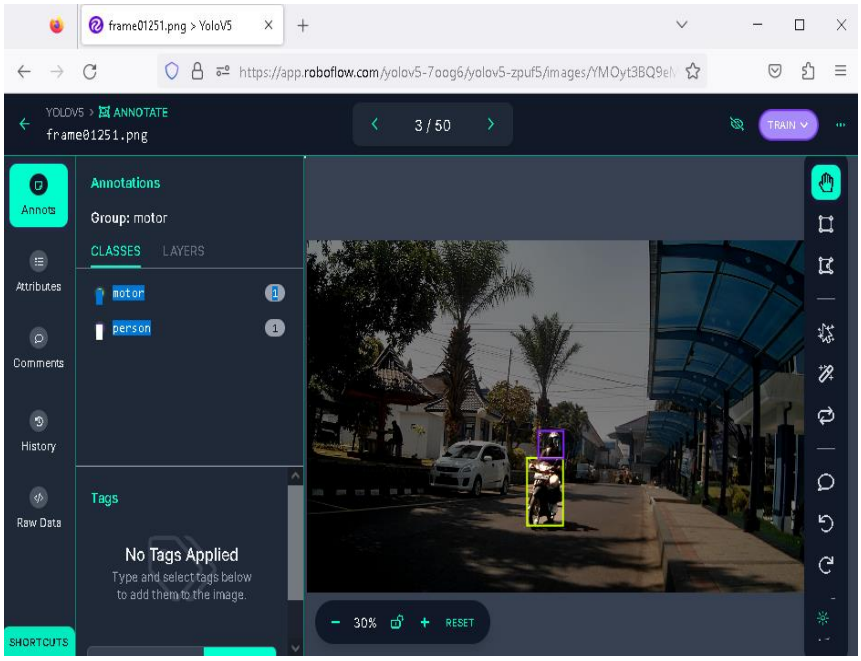

Fig. 5 Label Creation using Roboflow.

D. Data Augmentation

Data augmentation on images in the dataset is performed to enhance the variety of training data. Augmentation can involve changes in scale, rotation, cropping, flipping, color changes, and more. The goal of augmentation is to make the model more robust to variations in input data. Fig. 6 shows some examples of data augmentation on objects that can be performed using the Roboflow website platform.

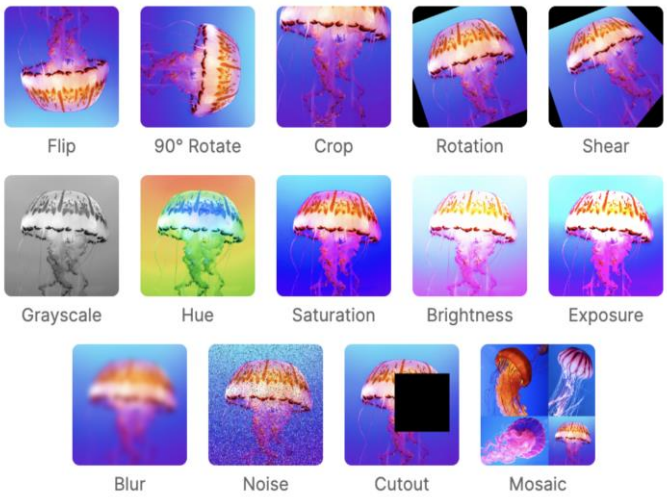

Fig. 6 Data Augmentation

III. RESULTS AND DISCUSSIONS

A. Training With Roboflow

The labeling procedure for each image is intended to differentiate between the various classes of objects for detection, specifically motorcycles and individuals. Following the labeling step, the images undergo data annotation, encompassing the provision of details about the objects through the inclusion of bounding boxes, class labels, and confidence scores. Following this, there is a resizing process to ensure uniform image dimensions. This process can bee seen in Fig. 7.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

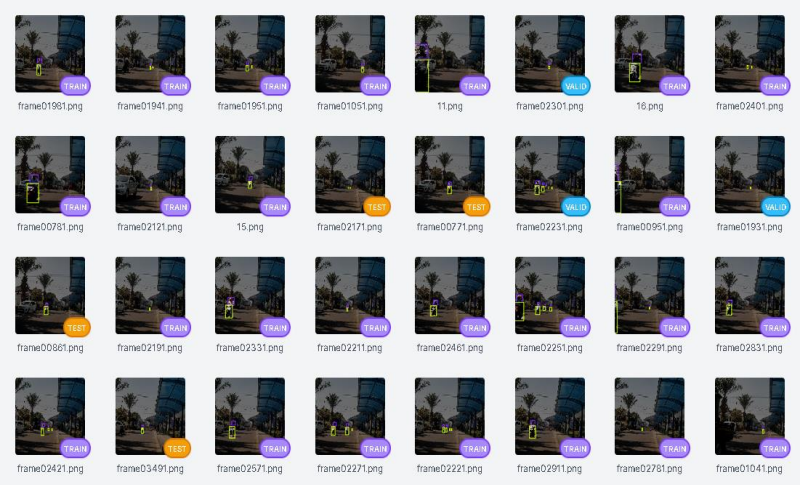

Fig. 7 Labelling Process

The labeled dataset will be processed for training by performing image and bounding box augmentation to generate variations in the dataset images used. Fig. 8 illustrates the augmentations used in this research. Subsequently, the dataset will be validated on the entire set of images by generating output versions, where Fig.9 represents the YOLOv5 PyTorch export process that produces a snippet of the project. This snippet is used as input data for Google Colab.

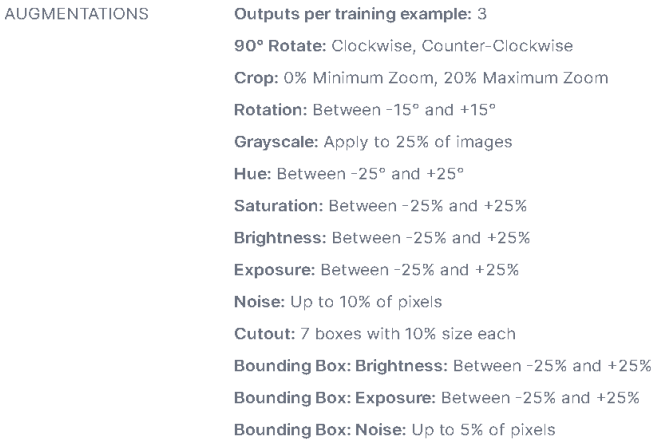

Fig. 8 Dataset Augmentation

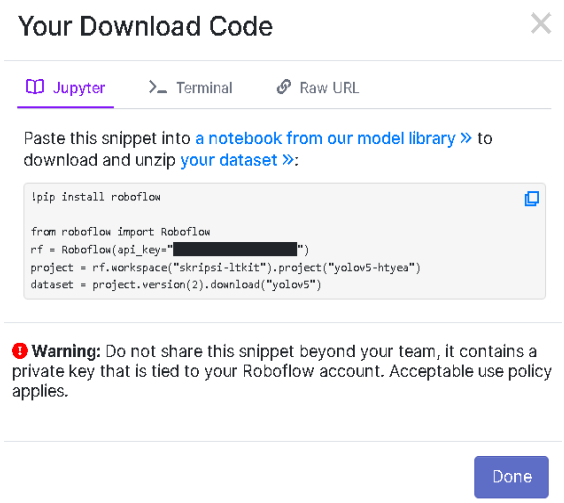

Fig. 8 Pytorch Export Process

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

The labeled dataset will be processed for training by performing image and bounding box augmentation to generate variations in the dataset images used. Image 8 illustrates the augmentations used in this research. Subsequently, the dataset will be validated on the entire set of images by generating output versions, where Fig.9 represents the YOLOv5 PyTorch export process that produces a snippet of the project. This snippet is used as input data for Google Colab.

B. Training With Google Colaboratory

This platform allows users to run Python scripts, import libraries, access GPU and TPU, and save and share code for free. Testing is performed by importing the required libraries and enabling the GPU. Furthermore, the Roboflow-trained dataset can be uploaded via snippets. Testing is conducted by varying epoch and batch size values using the YOLOv5n model to achieve good accuracy results.

C. Training Epoch

Epoch is a parameter used to determine how many times the system will repeat the training process using the dataset. Epoch represents the condition that describes the training updated in one complete cycle. This research involves multiple tests by varying the epoch values in the training process with values of 100 and 200. The goal of these tests is to determine the accuracy with respect to the mAP produced by epoch values. The test results for epoch values in Table 1 below show that the epoch testing was conducted twice with variations in epoch values, 100 and 200 by using different parameters.

TABLE I EPOCH TRAINING RESULTS

No	Image Size	Epoch	Batch	Model	mAP
1	640	100	16	YOLOV5N	90%
2	640	200	16	YOLOV5N	89%

Epoch values are a factor that can influence the best accuracy results of the system. The best results were obtained in the first experiment with an epoch value of 100. This is because the larger the epoch value, the better the results obtained. However, if there are too many epoch values, overfitting may occur, where the training data cannot generalize well to previously unseen data (test data or new data). As a result, the model will perform very well on the training data but less optimally on the test data. To address such issues, optimal testing of epoch values is performed by testing values that are not too large but still yield good accuracy results."

D. Testing Batch

The testing of batch size refers to the size of the batch (the number of machine learning or training instances used in one iteration). Testing different batch size values also affects the results obtained by the system. This research utilized a dataset consisting of 492 frames or images, and the tested batch sizes were 16 and 24 batches. The purpose of testing with different batch size values is to divide the dataset of 492 into a certain number of batches used in one training process, which can influence the accuracy of mAP. The following is the testing of batch size variations conducted in this research, as shown in Table 2. The results of the batch size testing are presented below:

TABLE II TESTING BATCH RESULTS

No	Image Size	Epoch	Batch	Model	mAP
1	640	100	16	YOLOV5N	87%
2	640	100	24	YOLOV5N	90%

Similar to epoch testing, batch value testing is conducted with 2 variations, 16 and 24. These values affect the data being tested. The best testing result obtained from batch value testing is in the second experiment when using 24 batches, resulting in an mAP value of 90%. Based on the obtained results, it can be seen that the mAP value undergoes a relatively small change when the batch value is varied, which is also influenced by the amount of data used. The testing of image datasets in one epoch consists of 21 iterations, obtained from dividing the total dataset by the batch value, resulting in 2100 iterations with an epoch of 100 times. The training results for batch 24 are better than batch 16 because more images are trained, providing a more stable accuracy outcome.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

E. Training Model YOLOv5n

Testing of the YOLOv5 model was also carried out during the training process. The YOLO model was tested in combination with several hyperparameters such as epoch and batch. The best testing results were then selected for use in real-time testing. Based on the testing of the YOLOv5n model, a mean average precision (mAP) of 0.9 was obtained. Image 11 represents the graph generated from testing the YOLOv5n model using parameters such as image size 640, epoch 100, and batch 24. Fig.9 shows several training and validation graphs, including box_loss, obj_loss, cls_loss, mAP_0.5, mAP_0.5:0.95, precision, and recall.Based on the box_loss, obj_loss, and cls_loss graphs, there is an exponential decrease, influenced by the number of epochs during training. The more epochs are run, the less loss will be generated for box and obj, but too many epochs are also not ideal for the system, as the decreasing loss indicates a negative exponential graph. Therefore, changes in loss when epochs increase are not very significant and only burden the system, consuming more time during the training process. Epoch 100 was chosen because it is the most optimal value and does not overly burden the system, and the resulting loss value is not too large. Moving on to the mAP 0.5, mAP 0.5:0.95, precision, and recall graphs, they are the opposite of the three previous graphs. As more epochs are trained, the values of the mAP, precision, and recall graphs will increase or approach 1.0. The larger the values obtained from these three graphs, the better the system is considered to be.

Fig.10 represents the result of the confusion matrix generated from the training and validation processes during the testing of the YOLOv5n model. The confusion matrix above illustrates the relationship between the predictive capabilities of each class and the training result values. The training results obtained when predicting an object still contain noise, such as background being detected when identifying an object, and vice versa when detecting an object, background is detected. The presence of noise is inevitable because during the labeling process, each class includes parts of the object and background, forming coordinate points that result in predicted values for a motor class with an average accuracy of 1 and predicting the person class with an average accuracy of 0.76.

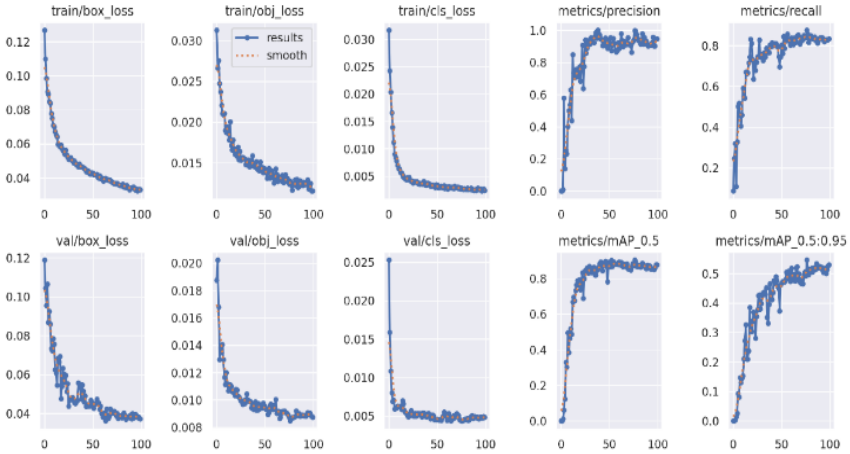

Fig. 9 Training Graph Results

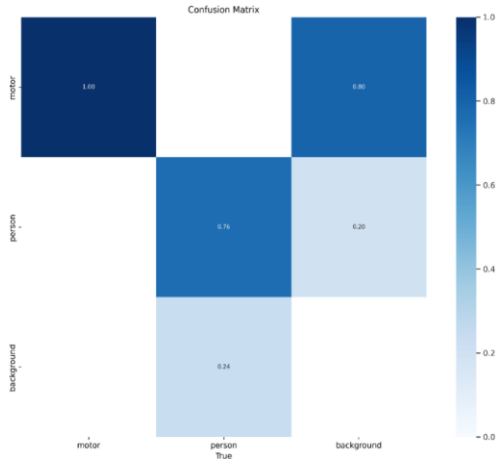

Fig. 10 Confusion Matrix

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

Fig. 11 represents the relationship between F1 score, precision and recall against confidence values. F1 Score providing an initial insight that the model has high confidence in its detections. Although high confidence can improve precision because the model will only perform detections with a high level of confidence, adjusting this threshold may make the model more selective in detections, potentially reducing recall. As a result, the resulting F1 score will reflect the balance between high precision and lower recall at the specified threshold level. Precision score can be observed in the generated graph that the classes of motor and person reach the highest precision values, affecting the confidence values. This indicates that the faster the motor and person classes reach their peak precision, the better the system is at predicting an object [14]. The confidence values reflect how well the system predicts an object.

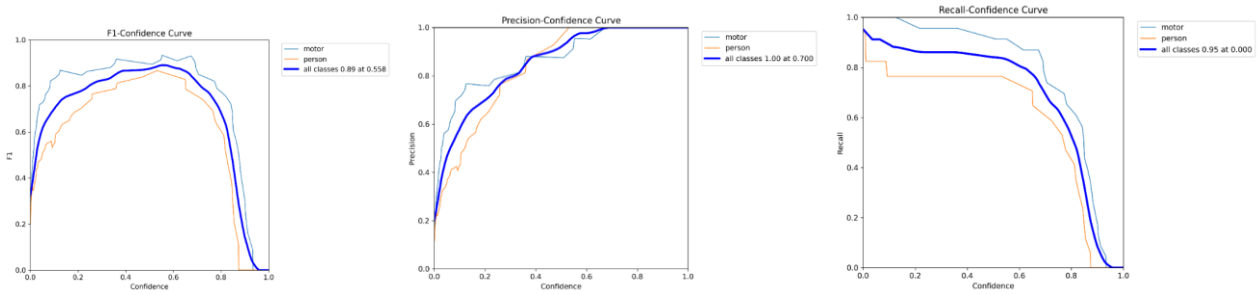

Fig. 11 Curve of F1 Score, Precision and Recall Against Confidence Values

F. Testing the Systems with Stationary and Moving Objects

The results of the system test that has been developed can be used to detect violations using dummy violation data simulated in a campus environment. In the testing process, two stages of data collection were conducted, varied by considering the influence of light intensity and distance, to understand the impact on the system's ability to detect violations. The first stage of testing was conducted with stationary objects, aiming to determine the maximum distance at which the system can detect violations. Subsequently, in the second stage of testing, an evaluation was performed on moving objects with the goal of testing the response and speed of information delivery. In this stage, the system was tested for its ability to detect and respond to quickly moving objects. The following graphs depict the tested system's capabilities through the two testing stages, as shown in Fig. 12.

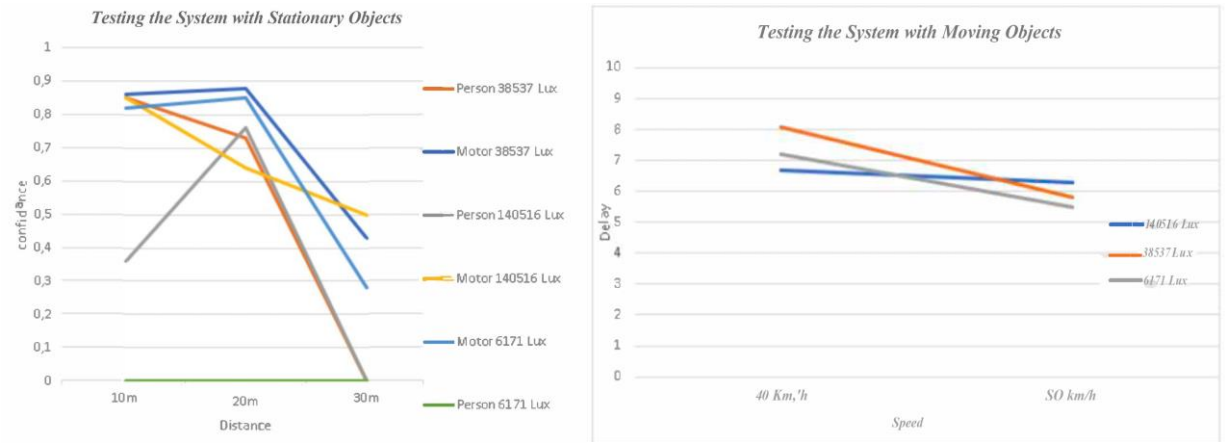

Fig. 12 Testing the Systems with Stationary and Moving Objects

Fig 12. shows the comparison of confidence values for each detected result using a webcam with different image qualities, influenced by differences in light intensity and distance to the object from the webcam. The system's ability to produce the best confidence values was observed with an intensity of 4440 Lux and an image quality that had a balanced brightness level, not too bright (high brightness) or too dark (low brightness). This factor affects the YOLOv5n system's ability to determine confidence values for detected objects. The detection test results for each object decrease with increasing distance and light intensity values, which can affect the image quality and result in blurriness on the object. The position of the object in the image also influences the detection results. If the object is closest to the center of the image, it has the largest image, resulting in the highest confidence values for each object.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 10, Issue 12, December 2023

DOI: 10.17148/IARJSET.2023.101201

Similarly, when the object is at the farthest position in the image, it has the smallest image. This is the reason for the undetection of the person object at a distance of 30m. The testing involved evaluating the speed of vehicles with the aim of determining how quickly the system could detect and send information via the Telegram application. The testing of response delivery speed showed differences in delays due to the influence of the object's speed. The speed at which objects pass by can affect how many images can be processed and the information delivery process by the system. The delay shown in the graph decreases, indicating that the faster the object passes, the faster the information delivery, as the system processes fewer images. The average delay produced by the YOLOv5n system with the total time for the entire test data can be seen in Table 3, it represents the time for delivering violation information when moving objects are detected by the YOLOv5n system.

TABLE III DELAY CALCULATION RESULTS

Speed and Lux	**Time/Packets**	**Delay**
40 km/h with 140516 *Lux*	$\frac{6{,}379\ s}{141\ packets}$	= 0,0452411348 s
50 km/h with 140516 *Lux*	$\frac{6{,}784\ s}{20\ packets}$	= 0,3397s
40 km/h with 38537 *Lux*	$\frac{8{,}112\ s}{108\ packets}$	= 0,0751111111 s
50 km/h with 38537 *Lux*	$\frac{5{,}825\ s}{19\ packets}$	= 0,3065789474 s
40 km/h with 6171 *Lux*	$\frac{5{,}337\ s}{129\ packets}$	= 0,0429224806 s
50 km/h with 6171 *Lux*	$\frac{7{,}246\ s}{33\ packets}$	= 0,2195757576 s
Total Average Delay		= 1,029129 s /6 times testing = 0,1715215 s = 171 ms

IV. CONCLUSION

The use of cameras to detect a moving object, especially on a webcam, can enhance a system using the YOLO algorithm as image processing to detect passing vehicles based on the classes to be detected. The YOLOv5 algorithm used in detecting two-wheeled vehicle violations with the collection of dummy data on road sections provides good results when using a high-resolution webcam (1920 pixels x 1088 pixels). The selected model for this test is YOLOv5n with 200 epochs and a batch size of 24.

This model was chosen because it has the lightest architecture that can be used with Raspberry Pi 4B, and training the dataset of 492 images with 4200 iterations resulted in an accuracy of 0.898 or 89.8% for all object classes. The results of testing light intensity with a range from 6171 Lux to 140516 Lux can affect the webcam's ability to produce video quality. The produced video can influence the YOLOv5n model's ability to detect violations at varying distances.

The YOLOv5n model can detect the entire object when the image is at its closest position to the webcam. However, if the object is at the farthest position, 30 meters away, the object is in the smallest image, and only motorcycle objects can be detected. The response in delivering violation information with an average delay of 171 milliseconds per data packet.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IARJSET.2023.101201

REFERENCES

[1]. S. R. Fadilah, H. Nishiuchi, and A. M. Ngoc, "The Impact of Traffic Information Provision and Prevailing Policy on the Route Choice Behavior of Motorcycles Based on the Stated Preference Experiment: A Preliminary Study," *Sustainability (Switzerland)*, vol. 14, no. 23, Dec. 2022, doi: 10.3390/su142315713.

[2]. Z. Li, Z. Huang, and J. Wang, "Association of Illegal Motorcyclist Behaviors and Injury Severity in Urban Motorcycle Crashes," *Sustainability (Switzerland)*, vol. 14, no. 21, Nov. 2022, doi: 10.3390/su142113923.

[3]. K. Sumit, K. Brijs, V. Ross, G. Wets, and R. A. C. Ruiter, "A Focus Group Study to Explore Risky Ridership among Young Motorcyclists in Manipal, India," *Safety*, vol. 8, no. 2, Jun. 2022, doi: 10.3390/safety8020040.

[4]. A. Haris, E. Yosrita, and R. A. Putra, "MODEL MONITORING DAN IDENTIFIKASI PELANGGAR DI JALUR TRANSJAKARTA MENGGUNAKAN LIBRARY TESSERACT OCR PADA RASPBERRY PI 3 MODEL B," 2017.

[5]. A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R. I. Taniguchi, "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow," *Sensors (Switzerland)*, vol. 19, no. 20, Oct. 2019, doi: 10.3390/s19204588.

[6]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." pp. 779–788, 2016. Accessed: Feb. 23, 2023. [Online]. Available: http://pjreddie.com/yolo/

[7]. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger." pp. 7263–7271, 2017. Accessed: Feb. 23, 2023. [Online]. Available: http://pjreddie.com/yolo9000/

[8]. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, doi: 10.48550/arxiv.1804.02767.

[9]. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, doi: 10.48550/arxiv.2004.10934.

[10]. W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2/FIGURES/5.

[11]. W. Wang *et al.*, "Efficient and Accurate Arbitrary-Shaped Text Detection With Pixel Aggregation Network." pp. 8440–8449, 2019.

[12]. Y. Egi, M. Hajyzadeh, and E. Eyceyurt, "Drone-Computer Communication Based Tomato Generative Organ Counting Model Using YOLO V5 and Deep-Sort," *Agriculture (Switzerland)*, vol. 12, no. 9, p. 1290, Sep. 2022, doi: 10.3390/AGRICULTURE12091290/S1.

[13]. N. Kevin Pratama and F. Tri Anggraeny, "DETEKSI LAMPU LALU LINTAS DAN ZEBRA CROSS MENGGUNAKAN MOBILENETV2 SINGLE SHOT DETECTOR," *Jurnal Informatika Kaputama (JIK)*, vol. 7, no. 2, pp. 225–232, Jul. 2023, doi: 10.59697/JIK.V7I2.140.

[14]. J. Yang, J. Chen, J. Li, S. Dai, and Y. He, "An Improved Median Filter Based on YOLOv5 Applied to Electrochemiluminescence Image Denoising," *Electronics (Switzerland)*, vol. 12, no. 7, Apr. 2023, doi: 10.3390/electronics12071544.

[15]. J. Zhang, J. Zhang, K. Zhou, Y. Zhang, H. Chen, and X. Yan, "An Improved YOLOv5-Based Underwater Object-Detection Framework," *Sensors 2023, Vol. 23, Page 3693*, vol. 23, no. 7, p. 3693, Apr. 2023, doi: 10.3390/S2307369

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License