Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 3 (2023) 715 - 721 ISSN Media Electronic: 2580-0760

Pedestrian Detection System using YOLOv5 for Advanced Driver Assistance System (ADAS)

Surya Michrandi Nasution[1], Fussy Mentari Dirgantara[2]
[1,2]School of Electrical Engineering, Telkom University
[1]michrandi@telkomuniversity.ac.id, [2]fussymentari@telkomuniversity.ac.id

Abstract

The technology in transportation is continuously developing due to reaching the self-driving vehicle. The need of detecting the situation around vehicles is a must to prevent accidents. It is not only limited to the conventional vehicle in which accident commonly happens, but also to the autonomous vehicle. In this paper, we proposed a detection system for recognizing pedestrians using a camera and minicomputer. The approach of pedestrian detection is applied using object detection method (YOLOv5) which is based on the Convolutional Neural Network. The model that we proposed in this paper is trained using numerous epochs to find the optimum training configuration for detecting pedestrians. The lowest value of object and bounding box loss is found when it is trained using 2000 epochs, but it needs at least 3 hours to build the model. Meanwhile, the optimum model's configuration is trained using 1000 epochs which has the biggest object (1.49 points) and moderate bounding box (1.5 points) loss reduction compared to the other number of epochs. This proposed system is implemented using Raspberry Pi4 and a monocular camera and it is only able to detect objects for 0.9 frames for each second. As further development, an advanced computing device is needed due to reach real-time pedestrian detection.

Keywords: pedestrian detection system; ADAS; intelligent transportation system; object detection; YOLOv5

1. Introduction

In the study of intelligent transportation, the existence of pedestrians is the most important thing that must be considered. The Indonesian Highway Capacity Manual [1] stated that, whenever pedestrians are crossing the roads, every motorized vehicle must be stopped and let them pass the roads. In a conventional vehicle, which is fully controlled by humans, there are lots of accidents caused by the foolishness of the drivers when facing the pedestrians who cross the roads [2].

Nowadays, the development in transportation technology allowed a computer to control a vehicle, commonly called an autonomous vehicle [3]. Based on this situation, the requirement of a pedestrian detection system is needed. It can work as an early warning system for pedestrians who crossed the roads. This system is needed, especially in Indonesia which had lots of pedestrians crossing the roads not at the right time and place [4].

The accidents occur not only in human-controlled vehicles, but also occurs in autonomous vehicles [5]–[7]. This condition happened because it is not reached the full stage of its autonomous system, furthermore, it still needs control from the drivers to prevent some unpredicted events (such as accidents) [8].

This paper proposed a simple solution for preventing the accident between vehicles and pedestrians, by using object detection and a minicomputer as the main processing unit. Our proposed framework contributes to prevents accident for pedestrians. For further purpose, we tried to reduce the number of accidents with vehicles and pedestrians. The proposed framework is the first compact dashcam that built to detect pedestrians to prevent the accidents. In road segments, the probability of pedestrians crossing the road is almost predictable. Its intention depends on the distance and speed of vehicles near pedestrians [9].

The pedestrians can be detected by using Convolutional Neural Network (CNN). This method applies a deep learning method that uses convolution by moving a convolutional multiplier [10]. Redmon, et al., improved the CNN due to fastening the object's detection result [11]. Our proposed system recognizes the pedestrians by using the fifth version of You Only Look Once (YOLOv5) which runs based on the CNN method [12].

Adopting YOLOv5 in a pedestrian detection system for an advanced driver assistance system (ADAS) has

Accepted: 20-01-2023 | Received in revised: 25-03-2023 | Published: 05-06-2023

several advantages, including high accuracy, real-time detection, a lightweight model, simplicity of training, and an open-source state. These benefits make it a viable option for enhancing the precision, speed, and effectiveness of pedestrian detection, which is crucial for ensuring the protection of drivers and pedestrians on the road [13].

As an object, the pedestrian has the standard feature of shapes, colors, and patterns [14]. The common methods used to detect pedestrians are Center and Scale Prediction (CNN-Based) [15], Improved XGBoost [16], Asymptotic Localization Fitting (SSD-Based) [17], Mask-Guided Attention Network [18], etc. This paper implements the YOLOv5 for detecting pedestrians using Raspberry Pi4 as the main processing unit.

As aforementioned, the pedestrian detection system is the most important part of the development of autonomous vehicles [19]. It can work as a warning system, such as speed control, brake recommendation, or other warnings through the Advanced Driver Assistance System (ADAS) [20].

The contents of this paper are structured as follows. The First section discussed the background of the pedestrian detection system using object detection. Section 2 discussed the proposed research methods for detecting pedestrians using YOLOv5 and Raspberry Pi4. Section 3 contains the result of the training model using various epochs and the detection result. At last, Section 4 shows the conclusions of this research.

2. Research Methods

In the autonomous vehicle, there are lots of sensors that are placed in order to be aware of the situation around it [21]. One of the ways to understand its nearby situation can be done by placing the cameras. It can capture images and later can be processed by detecting any object using computer vision techniques. This method allowed the computer to work like the eyes of humans by seeing every object that appears in front of the cameras [22].

The requirement analysis is built in order to create a pedestrian detection system. At least, it needs a standard resolution camera and an edge computer (Raspberry Pi4) which will be used as the system's main computing unit and object detection method for estimating the pedestrians in front of a vehicle. In general, this research consists of two stages. The first stage collects data on the pedestrian using the monocular camera, and the second stage detects pedestrians using YOLOv5.

Figure 1 shows the proposed method for detecting pedestrians crossing the streets. In the early stage, a monocular camera will capture an image or video which appeared from its point of view. Every image that is collected, will be preprocessed in order to simplify the images, easy to process, and adjust to the need of the convolutional stage.

Whenever the preprocessing step is done, the object detection process is begun. It tried to understand every object that appeared in an image. When it detects and recognizes any objects, a bounding box is drawn around the detected object. A bounding box has information such as its position and its size.

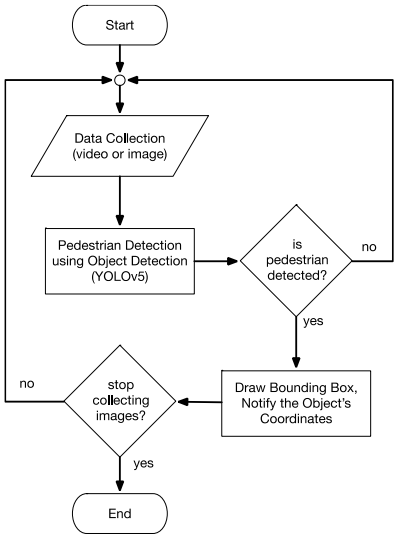

Figure 1. Proposed System

2.1 Data Collection with Monocular Camera

The pedestrian detection system proposed in this paper uses a minicomputer that is connected to a monocular camera. The definition of a monocular camera in this paper refers to a single camera which directed in line with the vehicle's direction.

The dataset used in this study is manually customized using the image search feature on search engines. The dataset is searched by tracing the shape of the pedestrian's form in accordance with the camera settings that have been carried out on the proposed system according to the needs of this study. The dataset distribution technique uses a ratio of 6:4. These are performed since that produces a somewhat more significant fraction of the training data, hence optimizing the training. The dataset of pedestrian image samples contains 21610 photos. Therefore, the quantity of training data utilized is 12966 photos, whereas the amount of test data utilized is 8644 images.

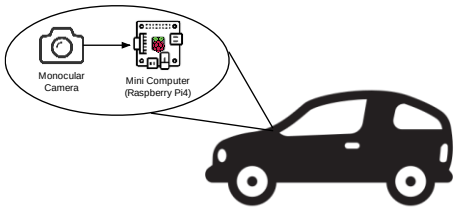

Figure 2. Pedestrian Detection System Placement Illustration

DOI: https://doi.org/10.29207/resti.v7i3.4884

The computer that is connected to the camera will process the images collected. It tried to collect all information or event that occurred in front of the vehicle. The illustration of the placement of the camera and minicomputer is shown in Figure 2.

In order to build the pedestrian detection system, there will be several requirements, namely the hardware and the development environment. Table 1 shows the requirement system used to build the pedestrian detection system.

Table 1. Requirement System

Pedestrian Detection System	Development Environment (Computer)
Raspberry Pi 4 (with installed Raspbian Buster OS) Monocular Camera LCD TFT 3.6” Storage System (microSD Card)	Operating System (minimum): Windows 10 / Mac OS 11.1) Processor: 2 GHz Quad-Core Intel Core i5 Memory: 16 GB Python 3.7, Library YOLOv5

As shown in Table 1, the Raspberry Pi4 works as the main computer placed in the vehicle. It runs with Raspbian Buster OS and receives images from a monocular camera. Meanwhile, at first, the pedestrian detection software is developed on a different computer, as specified in the table. Whenever it's ready to run, it will be customized and moved to the Raspberry Pi4 to detect any pedestrians.

2.2 Pedestrian Detection System

The pedestrian detection system is built based on the object detection method. One of the top methods for detecting objects is You Only Look Once (YOLO). In its development, the various version of YOLO is proposed by several researchers. Redmon et al. proposed the first version of YOLO in 2015 and It works based on the Convolutional Neural Network [11] to define objects. Figure 3 shows the architecture of YOLO which was developed by Redmon.

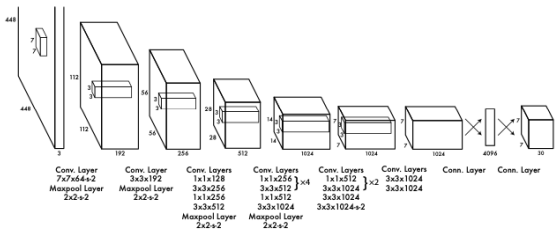

Figure 3. YOLO Architecture [11].

According to Deshpande et al. [23], YOLO is the fastest object detection method compared with Fast R-CNN and Faster R-CNN. Nowadays, YOLO is the most common method applied in intelligent transportation systems [24]–[26] or specifically in the development of autonomous vehicles [27].

Redmon et al., develop their methods to the next version which has an improvement in the accuracy and the speed of the algorithm [28]. In YOLOv3, the capability of detecting objects is improved [29]. It works for detecting common objects and it applies in every general case. YOLO's latest version is not developed directly by Redmon, but it is developed by other researchers in specified cases such as the intelligent transportation system [30]–[32].

To detect any pedestrians, the proposed system needs an image or video of the road’s situation. It needs another preprocessing step when the system received videos as input. The system will read and store all frames in the video input. Whenever the frames are stored, the objects in the computer will be detected immediately.

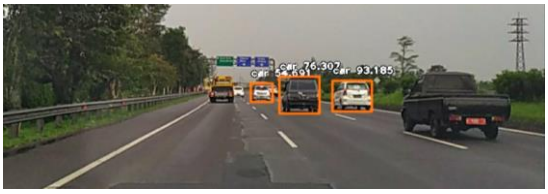

Figure 4. Object Detection in Dashcam (Smartphone) [33].

Figure 4 shows the illustration of the road's situation based on a simple dashcam (using a smartphone) proposed by Nasution, et al. [33]. It detects other vehicles which appeared on the roads. The pedestrian detection system that we proposed almost has a similar concept, but we focused on detecting pedestrians using a minicomputer and monocular camera.

2.3 The Scenario of Experiment

We simulate the pedestrian detection system using a minicomputer which is already attached by a monocular camera. In the simulation, we only tried to detect pedestrians who standing in front of the camera. As shown in Figure 5, the pedestrians are detected in the proposed system.

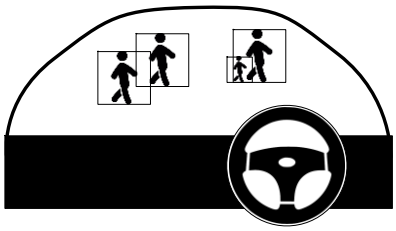

Figure 5. Pedestrian Detection System Simulation

In order to get a better detection system, there will be several model training with different numbers of epochs. According to Afaq and Rao [34], the definition of an epoch in machine learning is a process for one-time model training using the whole dataset. It tried to validate the training model by finding the minimum error among the various number of epochs. The loss function tried to find in this test is bounding box loss (Mean Squared Error) and object loss. There is no classification loss since the class that tried to detect is

DOI: https://doi.org/10.29207/resti.v7i3.4884

only pedestrians or persons. The Mean Squared Error (MSE) is calculated by using Equation 1.

$$MSE = \frac{\sum_{n=1}^{N}(X_n' - X_n)^2}{N} \quad (1)$$

X_n' and X_n are the predicted and actual values of the objects. The MSE value is determined based on the average gap between the predicted and actual values on N data.

Bounding box loss calculates by measuring differences between the predicted bounding box and the actual object's bounding box. It represents the capability of the method to find the center of an object and its bounding box to cover an object. Meanwhile, the object loss tends to show the confidence level of every detected object.

3. Results and Discussions

As mentioned in the previous section, in the beginning, we tried to find the minimum loss by training the model with various epochs. The loss function calculated in the experiment covers the bounding box and object loss. Later, we tried to detect the pedestrian by simulating our system outside the vehicle.

3.1 Training Model

The model is trained by using various epochs, namely 200, 500, 1000, and 2000 epochs. As aforementioned before, it is conducted in order to find the minimum loss with the most optimal number of epochs. The loss functions measured in the training model are bounding box and object loss. The classification loss is not measured in this paper since the category of object is limited to pedestrians (person).

Training Model with 200 Epochs*:* The 200 epochs training model takes time around 30 minutes with the average confidence level range of the detected object around 79%. In several epochs, there are some values of object loss that are less than 15%, which means the confidence level reached 85% for understanding the objects. The result of training for measuring the object loss with 200 epochs is shown in Figure 6.

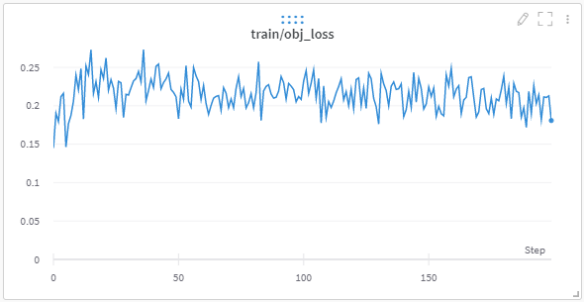

Figure 6. Result of Train Object Loss 200 Epochs

Meanwhile, Figure 7 shows the result of bounding box losses in the training with 200 epochs. The loss significantly dropped at the beginning of the training. At first, the loss reached 11% and it decreased to 9% at the 11th epoch. In the end, the loss reduction is less significant, and it reached 8.75%.

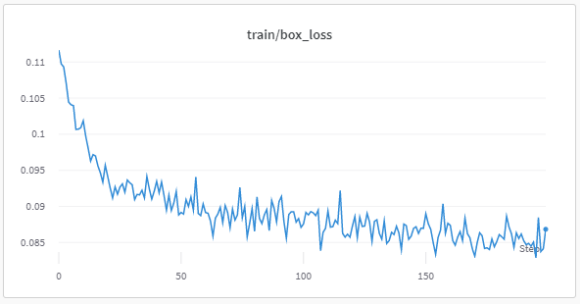

Figure 7. Result of Train Box Loss 200 Epochs

Training Model with 500 Epochs: The result of loss function testing in the training model is slightly better compared to the previous training. The object loss in detecting objects is around 21.69%, it has the same meaning as the confidence level in detecting objects is 78.31%. In this testing, the training model takes 75 minutes to finish its epochs. Figure 8 shows the training result of object loss with 500 epochs.

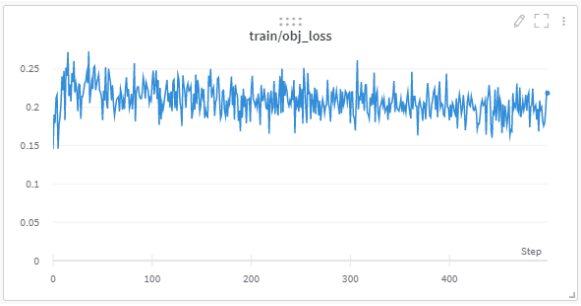

Figure 8. Result of Train Object Loss 500 Epochs

The testing result of bounding box loss in the model training with 500 epochs is better than the previous testing. Almost similar to the training model using 200 epochs, its loss reduced significantly at the beginning of epochs. In the end, the bounding box loss reached less than 0.08, as shown in Figure 9.

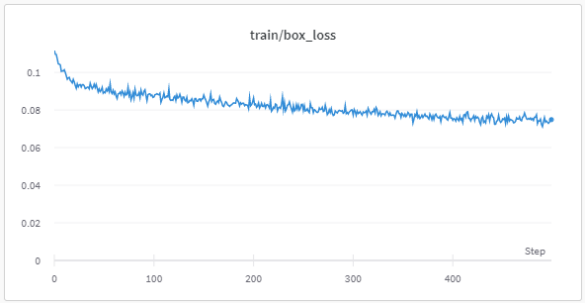

Figure 9. Result of Train Box Loss 500 Epochs

Training Model with 1000 Epochs: The loss in detecting objects when the model trained using 1000

DOI: https://doi.org/10.29207/resti.v7i3.4884

epochs tends to be greater than the previous model training. As shown in Figure 10, the average object loss is reaching 20.2%, which means the object is detected with a level of confidence is almost 80%.

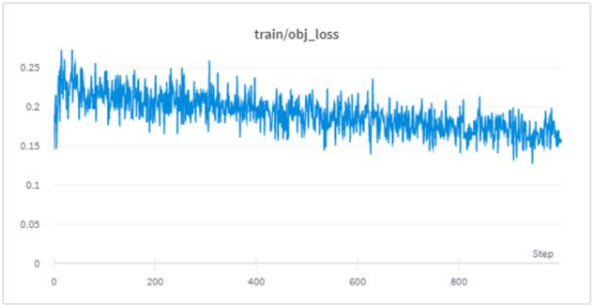

Figure 10. Result of Train Object Loss 1000 Epochs

On the other hand, the reduction of bounding box loss is not significant, compared to the previous model training with numerous epochs. Nevertheless, at the end of the epoch, the bounding box loss reached 0.06. It means the accuracy of the detected object's area reached 94%. Overall, the model that was trained using 1000 epochs takes 110 minutes to finish training as shown in figure 11.

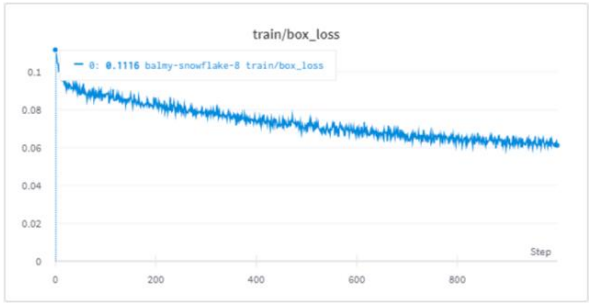

Figure 11. Result of Train Box Loss 1000 Epochs

Training Model with 2000 Epochs: The 2000 epochs model training delivers the best result among other training with numerous epochs. In Figure 12, at the first 200 epochs, the object loss is around 20.11% or the detected object's confidence almost reached 80%. At the time of the continuation of epochs in model training, the object loss is reduced and at the end of model training, it reached less than 15%.

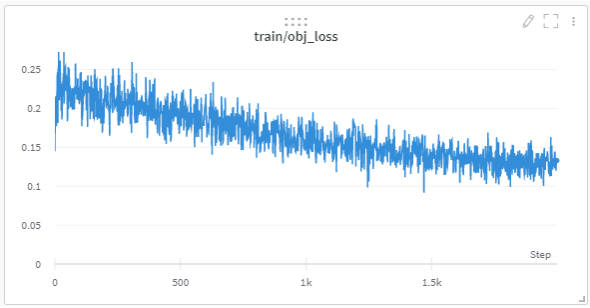

Figure 12. Result of Train Object Loss 2000 Epochs

As well as the object loss in this model training, the bounding box loss reached its lowest point. It reached 4% for its loss, and that means the system has 96% accuracy in the object's (pedestrian) detection area. Training with this number of epochs takes 180 minutes to build the model. Figure 13 shows the bounding box loss for model training with 2000 epochs.

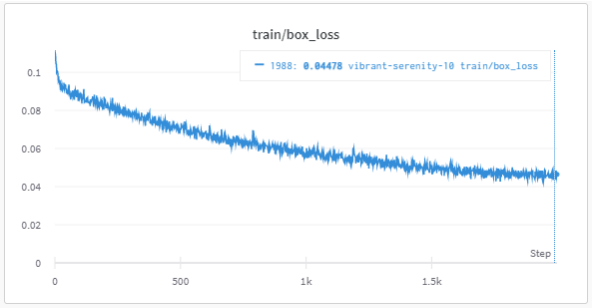

Figure 13. Result of Train Box Loss 2000 Epochs

Table 2 shows the comparison between training results based on numerous epochs. Based on this table, it can be concluded that the best epoch for modeling the pedestrian detection system could be trained by using 2000 epochs. It has the lowest object and bounding box loss between other numbers of epochs. Even though it has the minimum loss, it takes at least 3 hours to build the model.

Table 2. Model Training Result Comparison between Epochs

No. of Epoch	Obj. Loss (avg.)	Object Reduct -ion	B. Box Loss	B. Box Reduct -ion	Training Time (min.)
200	21.87%	-	8.75%	-	30
500	21.69%	0.18	7.5%	1.25	75
1000	20.2%	1.49	6%	1.5	110
2000	20.11%	0.09	4.48%	1.52	180

The most optimized number epoch, however, is around 1000. It not only has the maximum loss reduction (1.49 points) between epochs and a moderate reduction in measuring the loss of the bounding box (1.5 points) but also needs moderate training time to build a pedestrian detection model. As seen in Table 2, twice the number of epochs does not mean the loss is also reduced twice.

3.2 Discussions Pedestrian Detection Result

The testing of the pedestrian detection system is done by using Raspberry Pi which is integrated with a monocular camera. The system tried to detect the pedestrian using several samples of the recorded image as a testing method.

Figure 14 shows the result of the pedestrian detection system that has been built in this research. As seen in the figure, there is a pedestrian who stands in front of the view of the module (Raspberry Pi and Camera). Based on the testing result, it successfully detects a pedestrian at various distances. The pedestrian detection system also can recognize using live video

DOI: https://doi.org/10.29207/resti.v7i3.4884

streams, but it is only able to detect objects for 0.9 frames per second. The pedestrian system needs an improvement in the computing module to deliver real-time pedestrian detection.

According to the experiment that has been done, the pedestrian detection system using object detection (YOLOv5) works well for images. In order to detect objects using the camera's stream, the computer that is used must be improved due to speeding up the frame rate (near real-time detection). Whenever it is already able to detect objects in real-time, it can be relied on as a pedestrian detection system in autonomous vehicles.

Figure 14 Pedestrian Detection Results

4. Conclusion

Based on the testing results in this paper, we conclude the number of epochs in training a model can reduce the loss. But, expanding the number of epochs twice, does not mean the loss is reduced twice.

The maximum number of epochs that we tried is 2,000 epochs, and while it has an object loss of 20.11% and a bounding box loss of 4.48%, it takes at least three hours to develop a model. In the meanwhile, the optimal number of epochs is 1000 epochs; this has the highest possible object loss reduction (1.49 points) and only a moderate reduction in bounding box loss (1.5 points). In comparison to the many other epochs, the training time for this one looks to be quite short.

On the other hand, the implementation of the pedestrian detection system in Raspberry Pi as a main processing unit is only reliable for recorded images since it is only capable to deliver 0.9 frames/seconds. Due to the necessity of real-time detection which comes from the camera's stream, the main processing unit must be upgraded to better hardware such as jetson or any other minicomputer that is specified for object detection.

References

[1] *Manual Kapasitas Jalan Indonesia*. PT. Bina Karya (Persero), 1997.

[2] S. Razzaq, F. Riaz, T. Mehmood, and N. I. Ratyal, "Multi-Factors Based Road Accident Prevention System," *2016 Int. Conf. Comput. Electron. Electr. Eng. ICE Cube 2016 - Proc.*, pp. 190–195, 2016, doi: 10.1109/ICECUBE.2016.7495221.

[3] L. Liu *et al.*, "Computing Systems for Autonomous Driving: State of the Art and Challenges," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6469–6486, 2021, doi: 10.1109/JIOT.2020.3043716.

[4] R. Ahmad *et al.*, "Pedestrian User-Friendly Intelligent Crossing Advance For Improved Safety," *J. Keselam. Transp. Jalan (Indonesian J. Road Safety)*, vol. 9, no. 1, pp. 71–79, 2022, doi: 10.46447/ktj.v9i1.430.

[5] F. M. Favar, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining Accident Reports Involving Autonomous Vehicles in California.pdf," pp. 1–20, 2017.

[6] J. K. Gurney, *Sue My Car Not Me: Products Liability and Accidents Involving Autonomous Vehicles*, vol. 2013, no. 2. 2013.

[7] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: Disengagements, accidents and reaction times," *PLoS One*, vol. 11, no. 12, pp. 1–14, 2016, doi: 10.1371/journal.pone.0168054.

[8] M. Raza, "Autonomous Vehicles: Levels, Technologies, Impacts and Concerns," *Int. J. Appl. Eng. Res.*, vol. 13, no. 16, pp. 12710–12714, 2018, [Online]. Available: http://www.ripublication.com.

[9] S. Neogi, M. Hoy, K. Dang, H. Yu, and J. Dauwels, "Context Model for Pedestrian Intention Prediction Using Factored Latent-Dynamic Conditional Random Fields," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6821–6832, 2021, doi: 10.1109/TITS.2020.2995166.

[10] Y. Wang, B. Feng, and Y. Ding, "DSXplore: Optimizing convolutional neural networks via sliding-channel convolutions," *Proc. - 2021 IEEE 35th Int. Parallel Distrib. Process. Symp. IPDPS 2021*, pp. 619–628, 2021, doi: 10.1109/IPDPS49936.2021.00070.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only

DOI: https://doi.org/10.29207/resti.v7i3.4884

Look Once: Unified, Real-Time Object Detection," *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
[12] G. Jocher, "You Only Look Once v5," *YOLOv5*. .
[13] M. Sukkar, D. Kumar, and J. Sindha, "Real-Time Pedestrians Detection by YOLOv5," in *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*, 2021, pp. 1–6, doi: 10.1109/ICCCNT51525.2021.9579808.
[14] Z. Zhang, W. Tao, K. Sun, W. Hu, and L. Yao, "Pedestrian detection aided by fusion of binocular information," *Pattern Recognit.*, vol. 60, pp. 227–238, 2016, doi: 10.1016/j.patcog.2016.05.006.
[15] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 5182–5191, 2019, doi: 10.1109/CVPR.2019.00533.
[16] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters," *IEEE Access*, vol. 7, pp. 118310–118321, 2019, doi: 10.1109/ACCESS.2019.2936454.
[17] W. Liu, S. Liao, W. Hu, X. Liang, and X. Chen, "Learning efficient single-stage pedestrian detectors by asymptotic localization fitting," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11218 LNCS, pp. 643–659, 2018, doi: 10.1007/978-3-030-01264-9_38.
[18] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. S. Khan, and L. Shao, "Mask-guided attention network for occluded pedestrian detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 4966–4974, 2019, doi: 10.1109/ICCV.2019.00507.
[19] S. Iftikhar, Z. Zhang, M. Asim, A. Muthanna, A. Koucheryavy, and A. A. Abd El-Latif, "Deep Learning-Based Pedestrian Detection in Autonomous Vehicles: Substantial Issues and Challenges," *Electronics*, vol. 11, no. 21, p. 3551, 2022, doi: 10.3390/electronics11213551.
[20] H. S. Lee and K. Kim, "Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1652–1663, 2018, doi: 10.1109/TITS.2018.2801560.
[21] H. A. Ignatious, H. El Sayed, and M. Khan, "An overview of sensors in Autonomous Vehicles," *Procedia Comput. Sci.*, vol. 198, no. 2021, pp. 736–741, 2021, doi: 10.1016/j.procs.2021.12.315.
[22] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, p. 22, 2018, doi: 10.29099/ijair.v2i1.42.
[23] H. Deshpande, A. Singh, and H. Herunde, "Comparative analysis on YOLO object detection with OpenCV," *Int. J. Res. Ind. Eng.*, vol. 9, no. 1, pp. 46–64, 2020, doi: 10.22105/riej.2020.226863.1130.
[24] H. K. Jung and G. S. Choi, "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions," *Appl. Sci.*, vol. 12, no. 14, 2022, doi: 10.3390/app12147255.
[25] R. Dwiyanto, D. W. Widodo, and P. Kasih, "Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung," pp. 102–104, 2022.
[26] M. Kasper-Eulaers, N. Hahn, P. E. Kummervold, S. Berger, T. Sebulonsen, and Ø. Myrland, "Short communication: Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5," *Algorithms*, vol. 14, no. 4, 2021, doi: 10.3390/a14040114.
[27] W. Souidene Mseddi, M. A. Sedrine, and R. Attia, "YOLOv5 Based Visual Localization For Autonomous Vehicles," *Eur. Signal Process. Conf.*, vol. 2021-Augus, pp. 746–750, 2021, doi: 10.23919/EUSIPCO54536.2021.9616354.
[28] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
[29] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.
[30] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," *Sustain.*, vol. 14, no. 19, 2022, doi: 10.3390/su141912274.
[31] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," 2022, [Online]. Available: http://arxiv.org/abs/2209.02976.
[32] V. Pham, D. Nguyen, and C. Donan, "Road Damages Detection and Classification with YOLOv7," 2022, [Online]. Available: http://arxiv.org/abs/2211.00091.
[33] S. M. Nasution, E. Husni, Kuspriyanto, R. Yusuf, and R. Mulyawan, "Road Information Collector Using Smartphone for Measuring Road Width Based on Object and Lane Detection," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 2, pp. 42–61, Feb. 2020.
[34] S. Afaq and S. Rao, "Significance Of Epochs On Training A Neural Network," *Int. J. Sci. Technol. Res.*, vol. 9, no. 06, pp. 485–488, 2020, [Online]. Available: www.ijstr.org.

DOI: https://doi.org/10.29207/resti.v7i3.4884