

# Deep Learning for Pothole Detection on Indonesian Roadways

Hendra Kusumah<sup>1</sup>, Mohamad Riski Nurholik<sup>\*2</sup>, Catur Putri Riani<sup>3</sup>, Ilham Riyan Nur Rahman<sup>4</sup>

<sup>1</sup>Informatics Engineering Masters Program, Science and Technology Faculty, University of Raharja, Tangerang

<sup>2,3</sup>Information Systems Bachelor Program, Science and Technology Faculty, University of Raharja, Tangerang

<sup>4</sup>Informatics Engineering Bachelor Program, Science and Technology Faculty, University of Raharja, Tangerang

e-mail: [hendra.kusumah@raharja.info](mailto:hendra.kusumah@raharja.info), [nurholik@raharja.info](mailto:nurholik@raharja.info), [putri.riani@raharja.info](mailto:putri.riani@raharja.info), [ilham.riyan@raharja.info](mailto:ilham.riyan@raharja.info)

## Abstract

Accidents are common on Indonesian roadways. Accidents are caused by vehicles, motorcycles, and public transportation. Road fatalities are caused by speeding, alcohol, distraction, fatigue, and poor road conditions. There are numerous car accidents on Indonesian roadways. 30% of Indonesian traffic incidents are explained by road infrastructure and environmental conditions, 61% by driver skill and personality, and 9% by vehicle variables such as vehicle standardization. Cars are damaged, immobilized, and crashed as a result of road conditions. Every hour, three people pass away in traffic in Indonesia, according to authorities. According to the BPS's 2021 Land Transportation Statistics report, 31.91 percent of Indonesia's roads were damaged, totaling 174,298 kilometers. Accidents among Indonesian motorists are becoming more common as roads deteriorate. Using a single camera, a deep learning algorithm can recognize and detect road degradation such as potholes and road cracks. Train and process the model using transfer learning and fine-tuning on the Nano YOLOv5 model architecture. After being validated in three major scenarios, the model performs well with the appropriate confidence level. The precision metric for the model is 0.8, while recall and mAP:0.5 are both 0.5.

**Keywords**—Road Accidents, Deep Learning, Object Detection

## 1. INTRODUCTION

Each year, a significant number of road accidents occur in Indonesia [1]. These collisions encompass automobiles, motorcycles [2], trucks, and public transportation vehicles, among others [3], [4]. Road accidents are caused by reckless driving, excessive speeding, driving under the influence of alcohol or narcotics, driving while distracted, fatigue, and poor road conditions [5]. In Indonesia, poor road conditions contribute substantially to traffic accidents [6]. According to the director general of land transportation, Pudji Hartanto [7], 30% of traffic accidents in Indonesia are influenced by infrastructure and environmental conditions of vehicle paths, 61% by human factors, such as the ability to drive and the character of the driver while driving, and the remaining 9% by vehicle factors, such as the standardization of vehicles that are used properly or not on the streets. This poor road condition can have calamitous effects, such as increasing the likelihood of vehicle damage, decreasing the driver's ability to control the vehicle, preventing the vehicle from moving,

and increasing the likelihood of collisions between vehicles [8]. According to Indonesian police statistics, three people pass away every hour in road accidents [9]. According to the 2021 Land Transportation Statistics report by the Central Statistics Agency (BPS) [10], the cumulative length of all damaged roads in Indonesia in 2021 reached 174,291 kilometers, or approximately 31.91%. The extensive road damage in Indonesia has made drivers susceptible to collisions. Experts recommend studies that can help drivers become more aware of damaged roads, such as potholes and road fractures, which can lead to traffic accidents.

## 2. RESEARCH METHODOLOGY

### 2.1 Dataset

The dataset utilized in this study is derived from video logs that are publicly available on the YouTube site. The video-log (Vlog) format is employed, including footage of motorbike riders conducting MotoVlogs [11]–[14]. At least four videos are used as reference datasets in this research's training procedure, and three more videos are not processed in Roboflow with the intention of serving as test data. The movies classified as training data are then retrieved and processed using the Roboflow platform [15].



Figure 1 Image labeling sample



Figure 2 Image preprocessing and augmentation sample

The retrieved pictures are labeled in Roboflow using the boundingbox technique, as seen in figure 1 above. This seeks to make pothole identification easier by removing a broad backdrop, allowing the inference process to focus on only the damaged road parts. The quantity of data utilized after tagging was 952 images, which were then pre-processed and augmented by adding noise, exposure, and blur, as seen in Figure 2 above. Furthermore, picture resolution equalization is performed by converting all photos to a resolution of 480x480 pixels. The major goal of this preprocessing and augmentation is to generate a more diversified dataset so that new datasets may be obtained without re-labeling. The total number of images acquired is 2316 images [16], which are suitable for training and testing.

## 2.2 Model Architecture

Objects in pictures and videos are detected and located by computer vision. In real-world applications, deep learning models can recognize things reliably and effectively [17]–[19]. To recognize and locate objects in pictures and videos [20], [21], the object identification model employs a resilient backbone network and a detection head [22]. Data annotation [23], [24], optimization [25], and evaluation [26], [27] can train the model to recognize objects accurately and consistently. Architecture, loss function, and assessment criteria are determined by research aims and restrictions [28], [29].

There are two-stage and one-stage object detection models. Faster [30] and Mask R-CNN [31] are object detection models with two stages. Using an RPN, these models first suggest regions. Image object regions and bounding box coordinates are suggested by RPN. The second step categorizes and changes optimal zones for item identification. During region proposal, two-stage models focus on likely item locations and update predictions to increase detection accuracy. YOLO [32], [33] and SSD [34], [35] are object recognition models that estimate item bounding boxes and class probabilities in a single pass over an input image. These single-step models are fast and appropriate for real-time applications. They find items across the image by using anchor boxes of varied sizes and aspect ratios. One-stage models are faster than two-stage models, although they are less exact [36]–[38].

	from	n	params	module	arguments
0	-1	1	1760	models.common.Conv	[3, 16, 6, 2, 2]
1	-1	1	4872	models.common.Conv	[16, 32, 3, 2]
2	-1	1	4880	models.common.C3	[32, 32, 1]
3	-1	1	10560	models.common.Conv	[32, 64, 3, 2]
4	-1	2	29184	models.common.C3	[64, 64, 2]
5	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
6	-1	3	156028	models.common.C3	[128, 128, 3]
7	-1	1	221568	models.common.Conv	[128, 192, 3, 2]
8	-1	1	167944	models.common.C3	[192, 192, 1]
9	-1	1	442880	models.common.Conv	[192, 256, 3, 2]
10	-1	1	296448	models.common.C3	[256, 256, 1]
11	-1	1	164608	models.common.SPPF	[256, 256, 5]
12	-1	1	49536	models.common.Conv	[256, 192, 1, 1]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 8]	1	0	models.common.Concat	[1]
15	-1	1	263904	models.common.C3	[384, 192, 1, False]
16	-1	1	24032	models.common.Conv	[192, 128, 1, 1]
17	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
18	[-1, 6]	1	0	models.common.Concat	[1]
19	-1	1	90880	models.common.C3	[256, 128, 1, False]
20	-1	1	8320	models.common.Conv	[128, 64, 1, 1]
21	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
22	[-1, 4]	1	0	models.common.Concat	[1]
23	-1	1	12912	models.common.C3	[128, 64, 1, False]
24	-1	1	36992	models.common.Conv	[64, 64, 3, 2]
25	[-1, 20]	1	0	models.common.Concat	[1]
26	-1	1	74496	models.common.C3	[128, 128, 1, False]
27	-1	1	107712	models.common.Conv	[128, 128, 3, 2]
28	[-1, 16]	1	0	models.common.Concat	[1]
29	-1	1	179328	models.common.C3	[256, 192, 1, False]
30	-1	1	312160	models.common.Conv	[192, 192, 3, 2]
31	[-1, 12]	1	0	models.common.Concat	[1]
32	-1	1	329216	models.common.C3	[384, 256, 1, False]
33	[23, 26, 20, 32]	1	11502	models.yolo.Detect	[1, [[10, 27, 44, 48, 38, 94], [36, 68, 86, 152, 180, 137], [140, 301, 303, 264, 238, 542], [436, 615, 739, 386, 925, 792]], [64, 128, 192, 256]]

YOLOv5n6 summary: 281 layers, 3894312 parameters, 3894312 gradients, 4.3 GFLOPs

Figure 3 YOLOv5n architecture structure

The subject of this research is a one-stage model based on the YOLOv5 architecture [39]. Ultralytics provides a variety of YOLOv5 network version [40]. The network complexity of the YOLOv5 model is used to classify these models. The researchers chose the Nano version of the YOLOv5 architecture (YOLOv5n) for the study [41]. The network complexity of the YOLOv5 nano is exceedingly simple, as the term "nano" suggests. This is done so that the YOLO model can be run on low-resource platforms like smartphones [42], Raspberry Pi [43], and microcontrollers [44]. The architectural series used in this research model is depicted in the figure 3 above. As

illustrated, the model has 281 layers and a total of three million parameters.

### 2.3 Training Methodology

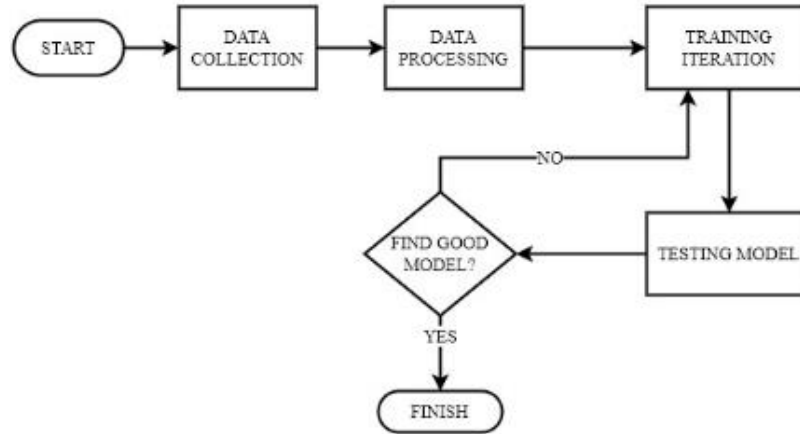


Figure 4 Research methodology flowchart

The flowchart in figure 4 above is an example of one type of research methodology offered by researchers. The research begins with the accumulation of data pertinent to the problems being encountered, particularly image data of potholes. After being collected, data on defects is utilized for testing and training purposes. In this investigation, the training procedure is divided into two stages: fine tuning and training. The purpose of both of these processes is to generate findings from trained models that meet the needs of the research. After the trained model has been produced, its performance is evaluated by applying it to video data not used during training.

## 3. RESULTS AND DISCUSSION

A Ryzen 9 CPU with 16 GB of RAM and a 4 GB Nvidia GeForce RTX 3050 graphics processor unit was utilized to conduct this investigation. In the widest sense, this research divided into three major categories: "fine tuning," "training," and "testing."

### 3.1 Fine Tuning

A transfer learning method is used in the fine tuning process. The pretrained model given by Ultralytics is used as the foundation model for training in this transfer learning approach. This pretrained model, however, continues to employ generic hyperparameters based on the supplied hyperparameters. As a result, researchers do fine tuning in order to adapt the hyperparameters to match the image dataset that being used. This fine tuning is accomplished by initiating the evolve process throughout the training phase, which updates the hyperparameters utilized at each training step. This evolve method seeks to identify the best hyperparameter configuration for the dataset in order for the model to better explore the dataset.

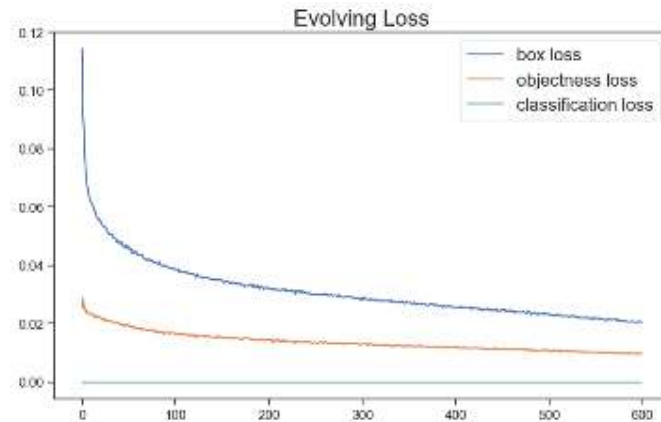


Figure 5 Fine tuning result

The fine tuning method took 800 iterations with a batch size of 32, an image size of 480x480 pixels, and the SGD (Stochastic Gradient Descent) optimizer. The data utilized in the fine tuning procedure is the same as the data used in subsequent training, namely the labeling and processing dataset. The figure 5 above are acquired after running the operation for several hours. Three loss functions are shown: box loss, objectness loss, and categorization loss. The term "box loss" refers to the model's bounding box loss. This loss function seeks to determine how accurate the bounding box is based on the model's prediction results for the item. Objectness loss is also used as a measure of how successfully the model recognizes an item in a image. Finally, classification loss is used to assess how accurately the model classifies an item depending on the class of objects presented. As can be seen in the figure 5 above, the YOLOv5n model is capable of minimizing its loss. Because the researcher only employed one class or one item detection in this investigation, the stable classification loss is 0. The goal of this single object detection is to lessen the strain on the model in inferring objects so that the resources required, in this case the laptop, are not overburdened. When examined through the loss graph in the figure 5 above, the outcomes of this fine tuning method are quite good, thus the researchers chose to apply the generated hyperparameters in this fine tuning process.

### 3. 2 Model Training

The training procedure comes next. The YOLOv5n6 model is trained using the same setup as during the fine tuning phase, as well as hyperparameters derived during the fine tuning process. This training procedure was repeated 1000 times with a patience configuration of 500 times. Following the completion of the training procedure, it was revealed that the training procedure had been terminated at the 753rd epoch. The training process is terminated since the model does not make substantial progress while training is conducted for the last 500 epochs, also known as the early stopping technique. This early stopping strategy is particularly effective in reducing excessive training time consumption, allowing the best model to be created with the least amount of time spent. The figure 6 below depicts the outcomes of this program.

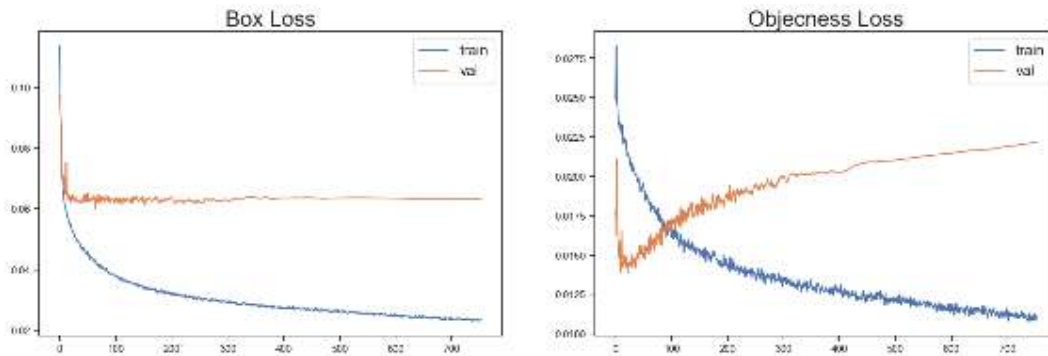


Figure 6 Training loss result

Based on the training results shown in the figure 6 above, the model can keep bounding box loss and objectness loss at their lowest levels. The model retains loss during the validation stage, as shown in the loss graph graphic figure 6 above. Despite the fact that its development on objectness loss appears to be going in the other way, the model is able to retain the consistency of the resultant loss. The extremely basic model complexity causes this circumstance, as the network processing model is not as detailed as the model with a larger complexity. However, as seen in the graph above, the model is still capable of maintaining loss at the lowest conceivable amount.

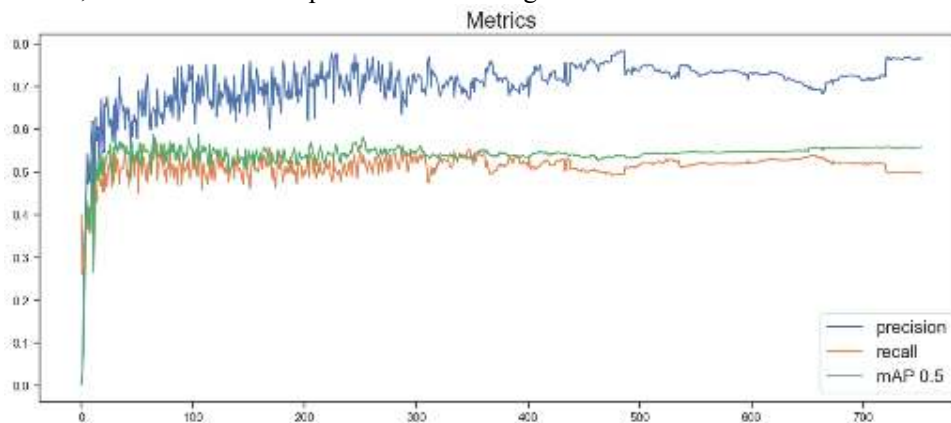


Figure 7 Training metrics history

In addition to loss graphs, researchers look at changes in accuracy, recall, and mean average precision (mAP), as seen in the figure 7 above. Since the beginning of training, the model has been able to reach the maximum degree of accuracy while still maintaining recall and mAP. The graph above shows that the model can detect pothole items with high precision and can detect things under specific situations. The ultimate outcome of this training method is a value of AuC (Area under Curves) of 0.56 from accuracy and recall.

### 3.3 Model Testing

At this stage, the model is evaluated with video data that was not included in the training dataset. When driving a motorized vehicle, such as a motorcycle, the video data is in the form of a video log (Vlog).



Figure 8 Testing inference result 1



Figure 9 Testing inference result 2

In general, the model can identify and detect potholes on paved roads. Figure 8 and 9 show that the model can detect minor and big potholes with motor speeds in the video log ranging from roughly 20-30 KM/hour. Furthermore, the researcher performed many situations that were used while testing conclusions. In this investigation, three primary scenarios were used, with confidence threshold values of 0.5, 0.6, and 0.8. The key difference between these three cases is the model's sensitivity to identifying things.

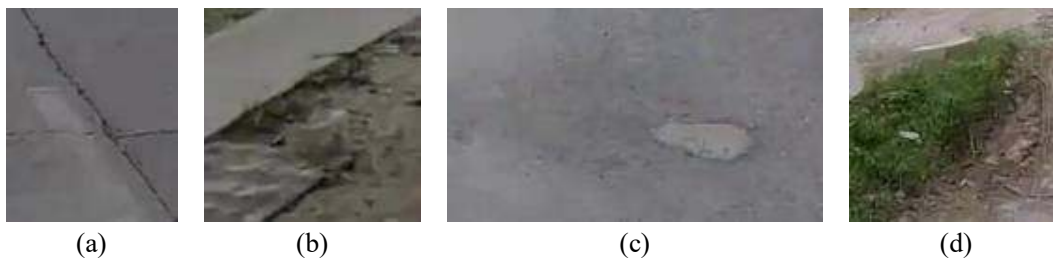


Figure 10 Inference result scenario 1

The model detects all items that are deemed potholes in the confidence threshold scenario of 0.5. However, at this level of sensitivity, the model recognizes items other than potholes, as shown in the figure 10 above. Figure 10 (a) shows how visible objects detect road joints and fractures. Furthermore, as illustrated in figure 10 (b), (c), and (d), the model recognizes road borders



and grass as well as asphalt areas on the road as potholes. Over-detection generates a pretty substantial bias in this image, prompting researchers to raise the confidence threshold to 0.6.

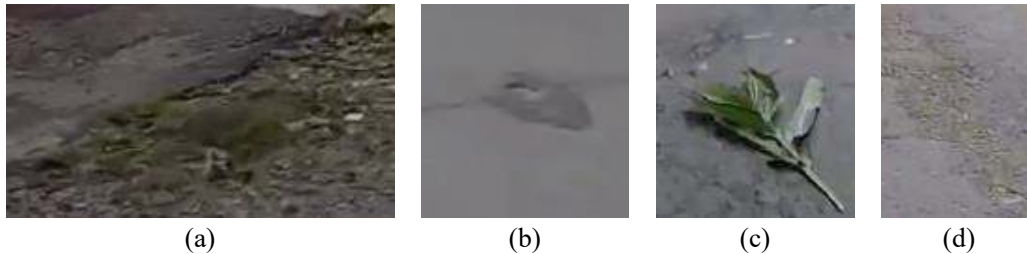


Figure 11 Inference result scenario 2

The model eliminates items better than a confident threshold of 0.5 in the scenario of a confident threshold of 0.6. However, as illustrated in figure 11 (a), the model still has difficulties eliminating things that appear to be damaged roadways but are not. Furthermore, as illustrated in figure 11 (b) and (d), the model detects asphalt patches that are relatively large in size as well as smooth holes. In this and earlier instances, the model still has difficulties removing things that are not road components, such as the tree trunks in figure 11 (c). Too much sensitivity leads to a significant bias during inference. Following multiple studies, the researchers opted to utilize a confident threshold value of 0.8 to avoid over-detection of things other than potholes.

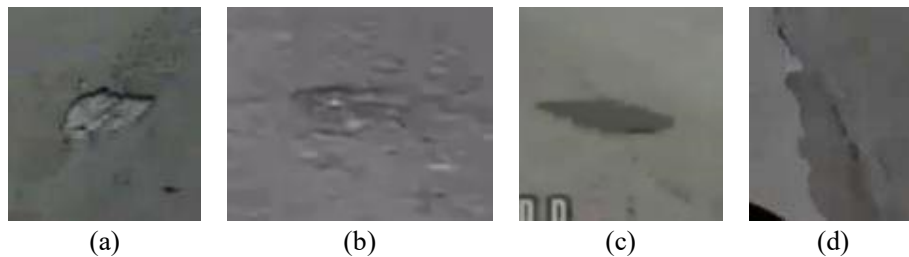


Figure 12 Inference result scenario 3

The model recognizes potholes with rather severe circumstances with this high enough confidence threshold, as seen in figure 12 (a) above. Furthermore, as demonstrated in figure 12 (b), the model is capable of identifying fine holes. However, as seen in figure 12 (c) and (d), the model still classifies asphalt patches as potholes as a result of this inference. The findings of the asphalt patch detection still cause a rather substantial bias in the inference results. This asphalt patch detection error can be caused by a number of factors, including a lack of accuracy in the dataset used during fine tuning and training, or the model still requires additional training to distinguish potholes from non-pothole objects. It is important to conduct additional study on this asphalt patch scenario in order for the model's inferences to be more precise and correct. In addition to detecting errors on asphalt, the model can detect potholes very well, and researchers believe the model is qualified to be implemented on edge computing devices such as Raspberry Pi, Arduino, or smartphones, which can later help motorists so that drivers are aware of potholes on the road, even though the model's implementation must be in several conditions, such as the confident threshold level, which must be adjusted to the speed of driving.



#### 4. CONCLUSION

This study focuses on the use of deep learning to the detection of potholes on roadways using only a single camera. The dataset used in this study is motoVlog video data collected from the YouTube site and processed with the Roboflow platform. The Nano version of YOLOv5 was employed in this work as the deep learning model architecture. This nano version of YOLOv5 is designed to run on devices with limited resources, such as smartphones, microcontrollers, and microprocessors. After 800 epochs of fine tuning and 753 epochs of training, the model with the highest efficiency was developed by combining transfer learning and fine tuning methodologies. The model can run well during testing, however the confidence threshold configuration needs to be modified in certain scenarios to reduce bias during inference.

#### 5. RECOMMENDATION

Researchers will do additional research by deploying models on edge computing devices. Furthermore, future research can construct model inferences on various vehicles (such as cars and motorcycles) with varying vehicle speed conditions. Also, the dataset utilized in the training process should be double-checked and focused on potholes that could cause traffic accidents. Data gathering, on the other hand, can be modified at various speeds, allowing for more precise and effective application on edge computing devices. With this advancement, it is believed that traffic accidents in Indonesia will be minimized and motorists will be more conscious of the road conditions they are driving on.

#### REFERENCES

- [1] “Angka Kecelakaan Lalu Lintas di Indonesia Meningkatkan di 2021, Tertinggi dari Kecelakaan Motor.” <https://databoks.katadata.co.id/datapublish/2022/03/24/angka-kecelakaan-lalu-lintas-di-indonesia-meningkat-di-2021-tertinggi-dari-kecelakaan-motor> (accessed May 26, 2023).
- [2] H. Widyastuti and L. Adriana, “ANALYSIS OF MOTORCYCLE TRAFFIC ACCIDENT BASED ON THE RIDERS’ CHARACTERISTICS IN KUPANG, EAST NUSA TENGGARA,” *Journal of Civil Engineering*, vol. 37, no. 1, pp. 23–26, Jun. 2022, doi: 10.12962/J20861206.V37I1.12087.
- [3] R. I. Idrus and Z. Khair, “ROAD USER AWARENESS AND ATTITUDE TO ROAD SAFETY IN INDONESIA 2000-2020,” *Jurnal Kemanusiaan*, vol. 19, no. 1, pp. 1–7, Jul. 2022, Accessed: May 26, 2023. [Online]. Available: <https://jurnalkemanusiaan.utm.my/index.php/kemanusiaan/article/view/410>
- [4] K. Fahmi and M. Arifin, “Characteristics of Traffic Accidents Frequency on Connection Road Between 3 Cities in Riau - Indonesia,” *Proceeding of International Conference on Science and Technology*, pp. 73–77, Dec. 2022, doi: 10.36378/INTERNATIONALCONFERENCEUNIKS.V0I0.2828.
- [5] F. Saffanah Didin, A. Widyanti, and H. Iridiastadi, “Exploring the Risk Factors for Rear-End Collisions on Indonesian Toll Roads”.
- [6] A. A. Anisarida, W. Santosa, and C. Sutandi, “TRANSPORTATION BUSINESS AND ROAD SAFETY MANAGEMENT IN INDONESIA,” *The Journal of Modern Project Management*, vol. 10, no. 1, pp. 270–283, 2022, Accessed: May 26, 2023. [Online]. Available: <https://journalmodernpm.com/manuscript/index.php/jmpm/article/view/555>

- [7] “31% Jalanan di Indonesia Rusak pada 2021.” <https://databoks.katadata.co.id/datapublish/2023/04/17/31-jalanan-di-indonesia-rusak-pada-2021> (accessed May 26, 2023).
- [8] “Accidents Caused By Poor Road Quality And Conditions In DC | Gilman & Bedigian.” <https://www.gilmanbedigian.com/accidents-caused-by-poor-road-quality-and-conditions-in-dc/> (accessed May 26, 2023).
- [9] “Kementerian Komunikasi dan Informatika.” [https://www.kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel\\_gpr](https://www.kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr) (accessed May 26, 2023).
- [10] “Kondisi Permukaan Jalan Nasional Tahun 2021 - Kondisi Permukaan Jalan Nasional | Open Data PUPR.” <https://data.pu.go.id/dataset/kondisi-permukaan-jalan-nasional/resource/797a1ab8-1afb-483c-8ecc-b8ceab49e7a4#view-graph:graphOptions:hooks:processOffset:bindEvents:graphOptions:hooks:processOffset:bindEvents:> (accessed May 26, 2023).
- [11] “Situasi Jalan Protokol Di Kota Tangerang H-2 Lebaran 2023 - YouTube.” [https://www.youtube.com/watch?v=zI6PaTfxN\\_0&t=14s](https://www.youtube.com/watch?v=zI6PaTfxN_0&t=14s) (accessed May 26, 2023).
- [12] “Jalan - Jalan Ke Perumnas Tangerang || Riding Perum Part 1 - YouTube.” <https://www.youtube.com/watch?v=4F6j4riMvXA&t=238s> (accessed May 26, 2023).
- [13] “HATI-HATI!! □ Jalan rusak di Kabupaten sukabumi | motovlog cewek #femalerider #motovlog #jalanrusak - YouTube.” <https://www.youtube.com/watch?v=OqT2IyeXjvU&t=296s> (accessed May 26, 2023).
- [14] “Motovlog melewati Jalan Raya Kali CBL Bekasi, Kondisi Rusak Parah..!! - YouTube.” [https://www.youtube.com/watch?v=7T\\_k7IXRS90&t=41s](https://www.youtube.com/watch?v=7T_k7IXRS90&t=41s) (accessed May 26, 2023).
- [15] “Roboflow: Give your software the power to see objects in images and video.” <https://roboflow.com/> (accessed Jan. 15, 2023).
- [16] “Pothole Motor Driving Object Detection Dataset by Muhammad Suzaki Zahran.” <https://universe.roboflow.com/muhammad-suzaki-zahran/pothole-motor-driving> (accessed May 26, 2023).
- [17] S. Tufféry, “Deep Learning for Computer Vision,” *Deep Learning*, pp. 347–430, Dec. 2022, doi: 10.1002/9781119845041.CH8.
- [18] D. Kesseli, V. Chidurala, R. Gooch, and G. Zhu, “A Combined Computer Vision and Deep Learning Approach for Rapid Drone-Based Optical Characterization of Parabolic Troughs,” *Journal of Solar Energy Engineering, Transactions of the ASME*, vol. 145, no. 2, Apr. 2023, doi: 10.1115/1.4055172/1145346.
- [19] L. Li, X. Zhang, J. Lian, Y. Zhang, and Y. Zhao, “Study on Practical Utility of Image Dehazing Algorithms Based on Deep Learning in Computer Vision Scene Understanding,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13605 LNAI, pp. 601–612, 2022, doi: 10.1007/978-3-031-20500-2\_49/COVER.
- [20] S. Belharbi, I. Ben Ayed, L. McCaffrey, and E. Granger, “TCAM: Temporal Class Activation Maps for Object Localization in Weakly-Labeled Unconstrained Videos.” pp. 137–146, 2023. Accessed: May 26, 2023. [Online]. Available: <https://github.com/sbelharbi/tcam-wsol-video>.
- [21] A. M. Roy, J. Bhaduri, T. Kumar, and K. Raj, “WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection,” *Ecol Inform*, vol. 75, p. 101919, Jul. 2023, doi: 10.1016/J.ECOINF.2022.101919.
- [22] B. Zhang, Z. Dong, J. Zhang, and H. Lin, “Functional network: A novel framework for interpretability of deep neural networks,” *Neurocomputing*, vol. 519, pp. 94–103, Jan. 2023, doi: 10.1016/J.NEUCOM.2022.11.035.

- [23] N. Mamat, M. F. Othman, R. Abdulghafor, A. A. Alwan, and Y. Gulzar, "Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach," *Sustainability* 2023, Vol. 15, Page 901, vol. 15, no. 2, p. 901, Jan. 2023, doi: 10.3390/SU15020901.
- [24] Q. Feng, K. He, H. Wen, C. Keskin, and Y. Ye, "Rethinking the Data Annotation Process for Multi-View 3D Pose Estimation With Active Learning and Self-Training," pp. 5695–5704, 2023.
- [25] H. J. Kim and M. K. Kim, "A novel deep learning-based forecasting model optimized by heuristic algorithm for energy management of microgrid," *Appl Energy*, vol. 332, p. 120525, Feb. 2023, doi: 10.1016/J.APENERGY.2022.120525.
- [26] R. Sadigov, E. Yildirim, B. Kocaçınar, F. Patlar Akbulut, and C. Catal, "Deep learning-based user experience evaluation in distance learning," *Cluster Comput*, pp. 1–13, Jan. 2023, doi: 10.1007/S10586-022-03918-3/TABLES/3.
- [27] P. J.; Sørensen *et al.*, "Evaluation of the HD-GLIO Deep Learning Algorithm for Brain Tumour Segmentation on Postoperative MRI," *Diagnostics* 2023, Vol. 13, Page 363, vol. 13, no. 3, p. 363, Jan. 2023, doi: 10.3390/DIAGNOSTICS13030363.
- [28] S. Shukla, N. Badal, and B. K. Thakur, "Introduction to Deep Learning," *Sustainable Computing*, pp. 253–267, 2023, doi: 10.1007/978-3-031-13577-4\_15.
- [29] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-Based Deep Learning," *Proceedings of the IEEE*, 2023, doi: 10.1109/JPROC.2023.3247480.
- [30] J. Xu, H. Ren, S. Cai, and X. Zhang, "An improved faster R-CNN algorithm for assisted detection of lung nodules," *Comput Biol Med*, vol. 153, p. 106470, Feb. 2023, doi: 10.1016/J.COMPBIOMED.2022.106470.
- [31] S. Fang, B. Zhang, and J. Hu, "Improved Mask R-CNN Multi-Target Detection and Segmentation for Autonomous Driving in Complex Scenes," *Sensors* 2023, Vol. 23, Page 3853, vol. 23, no. 8, p. 3853, Apr. 2023, doi: 10.3390/S23083853.
- [32] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognit Lett*, vol. 168, pp. 115–122, Apr. 2023, doi: 10.1016/J.PATREC.2023.03.009.
- [33] J. Zhang, Z. Ti, X. Lei, R. Arifando, S. Eto, and C. Wada, "Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses," *Applied Sciences* 2023, Vol. 13, Page 5802, vol. 13, no. 9, p. 5802, May 2023, doi: 10.3390/AP13095802.
- [34] R. Chen, J. Wu, Y. Peng, Z. Li, and H. Shang, "Solving floating pollution with deep learning: A novel SSD for floating objects based on continual unsupervised domain adaptation," *Eng Appl Artif Intell*, vol. 120, p. 105857, Apr. 2023, doi: 10.1016/J.ENGAPPAI.2023.105857.
- [35] H. Yang *et al.*, "PVT-SSD: Single-Stage 3D Object Detector With Point-Voxel Transformer." pp. 13476–13487, 2023.
- [36] D. Demetriou, P. Mavromatidis, P. M. Robert, H. Papadopoulos, M. F. Petrou, and D. Nicolaidis, "Real-Time Construction Demolition Waste Detection Using State-of-The-Art Deep Learning Methods; Single – Stage vs Two-Stage Detectors," *SSRN Electronic Journal*, 2023, doi: 10.2139/SSRN.4330569.
- [37] M. Yuzkat, H. O. Ilhan, and N. Aydin, "Detection of sperm cells by single-stage and two-stage deep object detectors," *Biomed Signal Process Control*, vol. 83, p. 104630, May 2023, doi: 10.1016/J.BSPC.2023.104630.
- [38] R. K. Al Romaihi, A. Belhi, and R. Hadjidj, "Evaluation of Single-Stage, Two-Stage, and Anchor-Free Object Recognition Methods for UAV Detection," *Lecture Notes in Networks and Systems*, vol. 578, pp. 775–783, 2023, doi: 10.1007/978-981-19-7660-5\_69/COVER.

- [39] “YOLOv5 - Ultralytics | Revolutionizing the World of Vision AI.” <https://ultralytics.com/yolov5> (accessed May 26, 2023).
- [40] D. DLUŽNEVSKIJ, P. STEFANOVIĆ, and S. RAMANAUSKAITE, “Investigation of yolov5 efficiency in iphone supported systems,” *Baltic Journal of Modern Computing*, vol. 9, no. 3, pp. 333–344, 2021, doi: 10.22364/BJMC.2021.9.3.07.
- [41] G. Jocher *et al.*, “ultralytics/yolov5: v6.0 - YOLOv5n ‘Nano’ models, Roboflow integration, TensorFlow export, OpenCV DNN support,” Oct. 2021, doi: 10.5281/ZENODO.5563715.
- [42] R. Hajar, P. Sejati, R. Mardhiyyah, N. Istiqomah, R. Imam, and B. Prasetya, “Real-time Smartphone Usage Surveillance System Based on YOLOv5,” *IJID (International Journal on Informatics for Development)*, vol. 11, no. 2, pp. 242–251, Feb. 2022, doi: 10.14421/IJID.2022.3766.
- [43] S. Li, S. Liu, Z. Cai, Y. Liu, G. Chen, and G. Tu, “TC-YOLOv5: rapid detection of floating debris on raspberry Pi 4B,” *J Real Time Image Process*, vol. 20, no. 2, pp. 1–13, Apr. 2023, doi: 10.1007/S11554-023-01265-Z/METRICS.
- [44] G. S. Ramya, “Smart Detection of Helmet Using IoT and Image Processing,” vol. 12, 2022.