JURNAL MULTIDISIPLIN MADANI (MUDIMA)

Homepage: https://journal.formosapublisher.org/index.php/mudima

ISSN: 2808-5639 (Online)

Research Article

Volume 3, No 6, June (2023)	DOI: https://doi.org/10.55927/mudima.v3i7.4931	Page: 1451–1460

Analysis of Helmet Detection on Motor Drivers to Detect Traffic Violations Using the You Only Look Once Method (Yolov4)

Nadia Hanifa Febriana

Universitas Pembangunan Nasional "Veteran" Jawa Timur

Corresponding Author: Nadia Hanifa Febriana nadiahf321@gmail.com

ARTICLE INFO

Keywords: YOLOv4, Deep Learning, Darknet, Breach, Crash

Received : *2 June*
Revised : *18 June*
Accepted : *19 July*

ABSTRACT

According to statistical data, the number of deaths due to accidents in Indonesia in 2017 was 30,568 people. Efforts are being made to reduce traffic violations, especially helmet violations. Helmets that must be worn by Indonesian motorcyclists must comply with the Indonesian National Standard (SNI), but there are still many non-SNI helmets circulating. A possible solution for monitoring is the identification of motorbikes in traffic based on Deep Learning. In this study, the classification of helmets was carried out using the YO-LO (You Only Look Once) method. The SNI helmet detection system aims to make drivers more disciplined in completing their riding equipment, especially helmets with SNI because this system requires riders to wear helmets that comply with LLAJ or SNI (Indonesian National Standard) helmets before riding. Trending Machine Learning and Deep Learning conduct research to discover new methods and advanced architectures such as YOLO (You Only Look Once). YOLO is an object detection network architecture that is claimed to be the "fastest deep learning object detector" that prioritizes accuracy and speed. With YOLOv4, violations by motorbike riders can be detected in real-time and whether the riders recorded on the camera are directly wearing SNI helmets, non-SNI helmets or not wearing helmets. The best accuracy for real-time motorcyclist violations with YOLOv4 is the best mAP value of 99.69%

INTRODUCTION

Based on data from the Central Bureau of Statistics, the number of victims who died in accidents in Indonesia was 30,568 in 2017. The percentage of the toll increases every year to 3.72% (BPS, 2017). The high number of victims who died was followed by the high number of violations that were often violated, one of which was not using a helmet (KORLANTAS, 2019). Various attempts have been made to reduce the number of traffic violations, especially violations of not using a helmet such as conducting outreach and raids on the highway. However, this solution has a weakness where limited human resources are inefficient in the long term.

Helmets that are mandatory for Indonesian motorcyclists should be helmets that comply with the Indonesian National Standard (SNI), but there are still many helmets circulating that do not comply. Helmet is part of motorized vehicle equipment that functions to protect the head in the event of a collision which is used to protect the public from possible serious head injuries due to motorcycle accidents and to improve motorcycle driving safety nationally. The government issues regulations through the Minister of Industry Regulation (Permen) No. 40/MINd/Per/6/2008 concerning the Application of Indonesian National Standards (SNI) for compulsory helmets for two-wheeled vehicles and Ministerial Regulation number No.40/MIND/PER/IV/2009 concerning delaying the implementation of SNI helmets must. Then reinforced Law number 22 of 2009 which requires all motorcyclists to wear helmets in accordance with SNI (Article 106, Paragraph 8). Helmets that comply with the Indonesian National Standard (SNI) are divided into 2 types, namely Open Face (Helmet shape that covers the head up to the neck and covers the front of the ears) and Full Face (Helmet shape that covers the top of the head, neck and back). mouth). And the helmet has a hard and smooth part which is the outermost part of the helmet and the inside which is installed to absorb impact energy, as well as the face of the helmet which can protect part or all of the face and is made of a clear layer. (BSN, 2010).

In overcoming these problems, the use of technology is needed to be able to monitor motorcyclists who commit violations. The branch of computer science that supports this system is computer vision. One solution that can be done for traffic monitoring is the recognition of motorbike objects in traffic based on the Deep Learning Journal of Informatics and Information Systems (JIFoSI) using one of the developments in the field of artificial intelligence, namely Computer Vision using object detection. on video surveillance cameras in public places. The computer must be able to recognize areas that are motorcycle objects in the image to make it easier for motorcycle users to detect helmets. The process of object recognition from an image is known as YOLO (You Only Look Once). It is hoped that this method can produce good accuracy and speed because the YOLO method is state-of-the-art in real-time object detection.

In this study, the classification of helmets was carried out using the YO-LO (You Only Look Once) method. YOLO uses a single Convolutional Neural Network (CNN) network for object classification and localization using bounding boxes. CNN is used to identify images and provide convincing results. The application of YOLO method has not been widely applied in image identification. This study implements the YOLO method as a helmet detector based on images. The method used is expected to be useful for detections that do not use helmets (Pramestya, 2018).

Because the use of SNI helmets is one of the requirements for safe riding for motorcyclists, however, it is still often ignored, which in turn causes the number of violations and accidents by motorcycle riders to be quite high. The SNI helmet detection system is expected to make drivers more disciplined in completing their riding equipment, especially SNI helmets, because this system forces riders to wear helmets that comply with the LLAJ Law or SNI helmets (Indonesian National Standards) before riding.

METHODS

1. Data Sharing

Split data is the process of dividing the collected dataset into the "SNI" dataset, "NoSNI" dataset and "NoHelm" dataset. The division of datasets based on this class will make it easier for researchers in the process of data augmentation, labeling and annotation of image.

2. Data Augmentation

Data augmentation is the process of changing or modifying an image so that the computer detects that the image being changed is a different image. Data augmentation is used to increase image data in the dataset. Image mirroring can be used as a way to reproduce image datasets. In addition to multiplying images, data augmentation can also change and exchange matrix values from an image. To simplify the data augmentation process, ImageMagick software is used to process images. ImageMagick is a software used for image processing of images.

3. Image Name Change

Changing the name of the image dataset needs to be done to make it easier to group and label data. Each class consists of 1000 images. Thus, the total number of images in the image dataset is 3000 images.

In the process of changing the image name, it takes a long time to change the name of 3000 images without spaces and parentheses. The way that can be done is to select all the images in the "SNI", "NoSNI" or "NoHelm" folders, then rename all the files according to the class name. The result will be {SNIJPG (1), SNIJPG (2), ..., SNIJPG (1000)}, {NoSNIJPG (1), NoSNIJPG (2), ..., NiSNIJPG (1000)} or {NoHelmJPG (1) , NoHelmJPG (2), ..., NoHelmJPG (1000)}. To simplify the process of changing the name according to the desired format, it is necessary to use batch file programs with the names "renamer-a.bat" and "renamer-b.bat". The program is a program made by the researcher himself, to simplify the process in setting the naming format as desired. The results obtained from the batch file program will be {SNIJPG1, SNIJPG2, ..., SNIJPG1000}, {NoSNIJPG1, NoSNIJPG2, ..., NoSNIJPG 1000} or {NoHelmJPG1, NoHelmJPG2, ..., NoHelmJPG1000}.

4. Image Annotation

The image dataset that has been divided based on the previous class, is then annotated on the images according to the YOLO format using LabelImg. LabelImg is a tool used for image labeling. LabelImg is written using Python and executed using QT for the graphical interface.

To carry out the image annotation process using LabelImg, researchers need to install the necessary tools. Next, all you have to do is run the LabelImg application and carry out the bounding box process or provide object boundaries to the image, then proceed with labeling the image based on class. After labeling, the annotation process is continued by saving the images in the YOLO format. The annotation results from the image will be saved in a text file (.txt) format. After all the data is annotated, combine the images in the "SNI", "NoSNI" and "NoHelm" classes in one folder with the name "obj".

5. Data Acquisition

The data acquisition system can be defined as a system that functions to retrieve, collect and prepare data, to process it to produce the desired data. The type and method chosen generally aim to simplify each step carried out in the entire process. A data acquisition system is generally written in such a way that the system functions to retrieve, collect and store data in a form that is ready for further processing.

The dataset has been annotated, then the process of dividing the images into training data and test data is carried out. The data to be trained and tested will be included in a text file (.txt). To simplify the data acquisition process quickly and automatically, a program for data acquisition is used in the form of a Python script, namely process.py.

6. Cloning and Building the Darknet

In this study, the flowchart in Figure 3.3 shows the workflow of the YOLOv4 algorithm. To start the data training process, the process of cloning and building Darknet is carried out and then stored in the specified folder. If you use Google Colaboratory as an application to run Python programming, Google Drive can be used as a data storage location.

It starts with mounting and linking Google Drive storage to Google Colaboratory. Make sure the folder matches the source code format used. After the mount and link process is successful, the next process is to clone Darknet into Google Drive storage. If Darknet cloning has been done before, then the cloning process will be considered a fatal error.

If you want to clone again, then delete the "darknet" folder on Google Drive storage. After the cloning process is successful, the next process is to change the folder directory to the darknet folder. Next, the process of building the Darknet can be done.

7. Loading Weight Pre-Trained YOLOv4 and Dataset

After the Darknet has been successfully built, add the dataset and text file resulting from the data acquisition to the "data" folder in Google Drive storage. The next process is a process that loads pre-trained weights. Pre-trained weights are required to become Transfer Learning. Transfer Learning consists of using the previously trained layers to build different networks that may have something in common with the first layer.

8. Hyperparameter Configuration

Hyperparameter configuration is a process used to define parameters when performing image training. YOLOv4 and YOLOv4-Tiny are hyperparameter configurations to be improvised. This hyperparameter configuration was taken based on journal references and previous research, then the researcher adjusted the number of iterations, convolutional filters and the number of classes according to the needs of the researcher. The following are the configuration parameters used.

Table 1. YOLOv4 Hyperparameter Configuration

Parameter Name	**Mark**
Number of Classes	3
Image dimensions	416x416
Max Iteration	6000
Number of convolution layers	110
Number of shortcut layers	23
Number of route layers	21
Parameter Name	5
The number of downsample layers	2
Number of upsample layers	3
Number of YOLO layers	1 x 1 and 3 x 3
Filter size	1 and 2
Stride on the convo layer	0.001
Learning rate	0.949
momentum	24

In table 1. the structure of CNN YOLOv4 is Darknet YOLOv4. Where this layer consists of 110 convolutional layers. In addition, there are 23 shortcut layers, 21 route layers, 5 downsample layers, 2 upsample layers, and 3 YOLO layers. Then after the hyperparameters are configured, the next process is parameter setting on YOLOv4-Tiny.

Table 2. YOLOv4-Tiny Hyperparameter Configuration

Parameter Name	**Mark**
Number of Classes	3
Image dimensions	416 x 416
Max Iteration	6000
Number of convolution layers	21
Number of route layers	11
Number of upsample layers	1
Number of YOLO layers	2
Filter size	1 x 1 and 3 x 3
Stride on the convo layer	1 and 2
Learning rate	0.00216
momentum	0.9
Convolutional filters	24

In Table 2, the structure of the CNN YOLOv4 architecture is Darknet YOLOv4. Where the layer consists of 21 convolutional layers. In addition, there are 11 route layers, 2 upsample layers, and 2 YOLO layers. After all, hyperparameters have been configured, the next process is setting up the model on the YOLOv4 architecture.

9. YOLOv4 Model Setup

YOLOv4 Model Tuning is the process of building the YOLOv4 architecture. The model used is Darknet-53. YOLOv4 uses a CSP connection with Darknet53 as the backbone for feature extraction. Darknet-53 architecture:

Table 3. Darknet-53 Architecture

Type		**Filters**	**Size**	**Output**
Convolutional		32	3 x 3	256 x 256
Convolutional		64	3 x 3 / 2	128 x 128
1x	Convolutional	3	1 x 1	
	Convolutional	32	3 x 3	
	Residual			128 x 128
Convolutional		128	3 x 3 / 2	64 x 64
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
Convolutional		256	3 x 3 / 2	32 x 32
	Convolutional	128	1 x 1	

3x	Convolutional	256	3 x 3	
	Residual			32 x 32
Convolutional		512	3 x 3 / 2	16 x 16
8x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			16 x 16
Convolutional		1024	3 x 3 / 2	8 x 8
4x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			8 x 8
Average Pool		Global		
Connected Softmax		1000		

The CSPDarknet53 model has higher accuracy in object detection compared to ResNet-based designs although ResNet has better classification performance. But the classification accuracy of CSPDarknet53 can be improved by Mish and other techniques.

10. Datasets Training

After configuring the hyperparameters and setting up the YOLOv4 model, the next step is the image training process from the pre-processed dataset. In this process, the dataset that will be used as training data and test data is as follows:

Table 4. Training Data

Total Image	Data Type
Training data and validation data	2700
Test data	300
Datasets (total)	3000

In the training, a process carried out, the learning rate of the hyperparameter configuration used is practically very small, namely 0.001. This causes the duration of the training process to last very long. However, this h affect a more accurate level of accuracy. Therefore, if an error occurs during the training process, the last data weights must be used to continue the training process, without having to start over again.

11. Saving Weights File and Loading Trained Weights

This process is important in the data training process. Where the data that has been trained will be stored in the form of a weights file (.weights). The process of saving the weights file starts in the first 1000 iterations, then it will be stored every 100 iterations as the last weights. Every multiple of 1000 iterations, will also be stored as a backup. If the max batches used are 6000, then there are 6 weights files, namely: iteration1000, iteration2000, iteration3000, iteration4000, iteration5000, iteration6000, last_weights, and best_weights.

Data that has been stored, of course, must be able to be loaded so that the data can be processed again. If an error occurs during training, these weights can be used to retrain the data.

12. Loading and Resizing Data Sets

This process is a testing process that is carried out when the training process is in progress. In this process, the test data will be loaded and resized to 416 × 416 image dimensions. In this process, the mAP (mean Average Precision) results, average loss, and the time required in the training process will also be rearranged. If the latest mAP results or accuracy are higher than the previous results, then these results will be stored as the best weights. Conversely, if the latest mAP results or accuracy are lower than the previous results, then these results will be stored as the last weights.

13. Data Evaluation

After data training has been successfully carried out, the next process is the data evaluation process. The following are the stages of data evaluation:

12. Calculating Image Detection

In this process, the detected image and the total object will be calculated. In this process, in addition to the number of detected objects, the AP value, the number of T (True), the number of F (False) and the number of unique truth counts will also be calculated. In the image detection calculation process, the results of correct detection and false detection obtained will determine the level of accuracy of image detection.

13. Storing Output Results and Calculating Performance Parameters

Based on the detection results and image detection calculations during the research process, the test results data will be stored in the backup. This storage process is useful to be able to know the progress of the increase in accuracy obtained.

Calculation of performance parameters needs to be done to find out the process of increasing or decreasing accuracy. This process can provide temporary conclusions, whether the level of accuracy obtained is accurate for use or not. Accuracy results are displayed in graphical form. However, there is an obstacle during the process of displaying accuracy graphs, namely the program cannot display the performance that has been passed before if there are problems during data training. So, the function from the last weights file cannot be displayed.

14. Research Scenario

The traffic violation detection research scenario that was carried out was to compare the accuracy results based on improvised parameter configurations between YOLOv4 and YOLOv4-Tiny. Steps to compare the accuracy of mAP values on the same image. There are 2 research scenarios, in the first scenario, the selfie image used is an image using a helmet according to standards (SNI), an image that does not use a helmet according to standards (NoSNI) and does not use a helmet (NoHelm) and a photo of the helmet of the researcher and his colleagues. While the second scenario, the image used for detection is a picture of a helmet of someone who is riding a motorbike.

15. Test Scheme

In the testing scheme, the first step is to prepare the objects to be investigated, namely SNI helmets, NoSNI helmets and NoHelm and Google Collaboratory to carry out detection. Then, gather colleagues who will be photographed directly. The test scheme is a study of selfies and driving images of the researcher and his colleagues. Where the selfie image data needed is 5 photos of helmet users (researchers and colleagues) while using helmets according to standards (SNI), 5 photos (researchers and colleagues) who do not use non-standard helmets (NoSNI), 5 photos of helmet users (researchers) and colleagues) are not wearing a helmet (NoHelm). Images were taken during the day (light) and night (dark). Then when taking photos, pay attention to the lighting, distance and angle of taking the photo. The selfie and driving images are entered into Google Drive first before running the detection. Furthermore, after all the files are ready, the program is ready to run and detect the results of the data that has been retrieved.

RESULTS AND DISCUSSION

Results

After knowing the methodology used for research on violation detection on motorists, the next step is to apply the research method and display data that supports the results of the research conducted. The results of applying the research method are divided into 4 parts, namely: 1. Results of Pre-Processing Data, 2. Results of Data Training, 3. Results of Data Evaluation, and 4. Results of Driver Violation Detection.

a. Results of Pre-Processing Data

In the data pre-processing process, the image dataset that has been obtained from the Kaggle open source is processed according to the YOLO format. In the data pre-processing process, there are 5 processes carried out, namely: 1. Data sharing, 2. Augmentation, 3. Name change, 4. Data annotation, 5. Data acquisition.

b. Image Name Change

After carrying out the image augmentation process, the next step is the process of changing the name of the image. This process is needed to simplify the research process and make image datasets more well-organized. On Windows OS, you can change the name by selecting all the image data in the folder, then changing the name of the data according to the class. The result will be {SNIJPG (1), SNIJPG (2), ... SNIJPG (1000)}, {NoSNIJPG (1), NoSNIJPG (2), ...NoSNIJPG (1000)}, or {NoHelmJPG (1)), NoHelmJPG (2), ... NoHelmJPG (1000)}. However, because the required image data naming format is without spaces and brackets, the researcher created a "renamer" program with a batch file format, to remove spaces and brackets. This program can only run on Windows OS.

c. Image Annotation

In this process, the image dataset will be annotated according to the YOLO format. The step taken is to run the LabelImg program in the image dataset folder that will be annotated. In this process, enter one of the class folders (eg the "NoHelm" folder) and run the program. After successfully running, do a bounding box on the object (rider) you want to detect. After that, label the image according to the class.

After carrying out the bounding box and labeling processes, the next step is to annotate the data according to the YOLO format. The data will be stored in the form of a text file (.txt) for each image that has been bounded boxed and labeled.

d. Data Acquisition

After the dataset has been fully annotated, then the process of dividing the images into training data and test data is carried out. The data to be trained and tested will be included in a text file (.txt). To simplify the data acquisition process quickly and automatically, a program for data acquisition is used in the form of a Python script, namely process.py.

If the program is successfully executed, the program will create text files, namely train.txt and test.txt. the data is the process of selecting images to be used as training data and images to be used as test data. The training data contains 90% of the dataset's images, while the test data contains 10% of the dataset. So there are 2700 images for training data and 300 images for test data. Next, is the process of merging the data, which was initially divided into "SNI", "NoSNI" and "NoHelm" folders, then the image data is combined into one folder, namely the "obj" folder.

e. Data Training Results

If the program is successfully executed, the program will create text files, namely train.txt and test.txt. the data is the process of selecting images to be used as training data and images to be used as test data. The training data contains 90% of the dataset's images, while the test data contains 10% of the dataset. So there are 2700 images for training data and 300 images for test data. Next, is the process of
mAfter doing the virgin preparation in the data preprocessing process, the next process is data training. Before training the image dataset, the YOLO architecture that will be used needs to be configured first, so that the training process can run properly and optimally.
merging the data, which was initially divided into "SNI", "NoSNI" and "NoHelm" folders, then the image data is combined into one folder, namely the "obj" folder.

While the folder mount process is running, permit access to Google Colaboratory to use Google Drive storage.

f. Data Evaluation Results

The data evaluation process can take place during the data training process, which is displayed for the first time in the 1000th iteration. Furthermore, the evaluation of the best accuracy level and the last accuracy level will be displayed every 112 iterations. However, the results are presented briefly and the training process is still ongoing, so one can only see the evaluation at a glance.

Discussion

Based on the research results that have been obtained, the following is a discussion of the results obtained from the research that has been conducted:

a. Discussion of Pre-Processing Data Results

In the data pre-processing process, the processing method used is quite complicated and requires quite a long time. The data-sharing process is fairly easy because it only divides the image dataset obtained into 3 folders, namely "SNI", "NoSNI" and "NoHelm".

In the data augmentation process, the time needed for data augmentation using ImageMagick is faster than manual augmentation. Because ImageMagick can perform image augmentation using the command prompt. The name change process is also faster and can be arranged according to the desired naming, namely without spaces and without parentheses. The batch file program created by researchers can change file names quickly.

b. Discussion of Data Training Results

In the data training process, the YOLO architecture development process for data training is quite simple to build. However, the data training process using the YOLOv4 configuration takes a very long time. Because in the data training process, the YOLOv4 configuration used has a small learning rate. That is 0.001. In addition, the data training process has a maximum iteration limit of up to 6000 batches. This causes the data training process to be very long. It takes approximately 16 hours for data training. Whereas in the data training process using YOLOv4-Tiny, the time needed is very short. YOLOv40Tiny training process, less than 2 hours. Because in the data training process, the YOLOv4-Tiny configuration used has a higher learning rate than YOLOv4, which is 0.00216.

CONCLUSION

Analysis of Helmet Detection on Motorcyclists To Detect Traffic Violations Using the You Only Look Once (YOLOv4) Method which has been successfully carried out using the YOLOv4 architecture. Based on the results of the analysis above, it can be concluded several conclusions as follows:

1. The research process was carried out very well and the results were as expected. The YOLOv4 and YOLOv4-Tiny parameter configuration implementations used affect the data training process for object detection. Improving the YOLOv4 parameter configuration with a learning rate of 0.001 for data training required a very long time (approximately 16 hours), but the training process and data evaluation went quite well with a 75% mAP value. Whereas in the YOLOv4-Tiny parameter configuration improvisation with a learning rate of 0.00261 for data training, the mapped value obtained was still lower than YOLOv4. The time required for the YOLOv4-Tiny configuration for the data training process is also shorter than the YOLOv4 configuration
2. The best mAP value obtained in the YOLOv4 parameter configuration is 99.69%. While the best mAP value obtained in the YOLOv4-Tiny parameter configuration is 62.5%. The mapped value obtained on YOLOv4 is fully feasible for real testing. Meanwhile, YOLOv4-Tiny is not feasible for real testing.

REFERENCES

Bisong, E. (2019). 10. Apress, Berkeley, CA. https://doi.org/https://doi.org/10.1007/978-1-4842-4470-8_7

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. http://arxiv.org/abs/2004.10934

Cui, Zhe and Sun, Hong-Mei and Yin, Ruo-Nan and Gao, Li and Sun, Hai-Bin and Jia, R.-S. (2021). No Title. In Real-time detection method of driver fatigue state based on deep learning of face video.

Fikriya, Z. A., Irawan, M. I., & Soetrisno., S. (2017). Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital. Jurnal Sains dan Seni ITS, 6(1). https://doi.org/10.12962/j23373520.v6i1.21754

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193–202. https://doi.org/10.1007/BF00344251

Giancini, D., Puspaningrum, E. Y., & Via, Y. V. (2020). Identifikasi Penggunaan Masker Menggunakan Algoritma CNN YOLOv3-Tiny. Prosiding Seminar Nasional Informatika Bela Negara, 1, 153–159. https://doi.org/10.33005/santika.v1i0.41

Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language). Memahami Penggunaan UML (Unified Modelling Language), 6(1), 1–15. https://informatikamulawarman.files.wordpress.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf

Kadir, A. (2013). Buku pintar programmer pemula php.

Khairunnas, K., Yuniarno, E. M., & Zaini, A. (2021). Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot.

Jurnal Teknik ITS, 10(1). https://doi.org/10.12962/j23373539.v10i1.61622

Kusuma, T. A. A. H., Usman, K., & Saidah, S. (2021). People Counting for Public Transportations Using You Only Look Once Method. Jurnal Teknik Informatika (Jutif), 2(1), 57–66. https://doi.org/10.20884/1.jutif.2021.2.2.77

Munantri, N. Z., Sofyan, H., & Florestiyanto, M. Y. (2020). Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon. Telematika, 16(2), 97. https://doi.org/10.31315/telematika.v16i2.3183

Nugroho, A. (2010). Rekayasa Perangkat Lunak Menggunakan UML & Jav. Yogyakarta :Andi, 2009. http://laser.umm.ac.id/catalog-detail-copy/130001892/

Perkovic, L. (2012). Introduction to Computing Using Python: An Application Development Focus.

Pramestya, R. H. (2018). Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO Berbasis Citra Digital. Institut Teknolgi Sepuluh Nopember, 91. http://repository.its.ac.id/59044/1/06111650010019-Master_Thesis.pdf

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Computer Science > Computer Vision and Pattern Recognition. https://doi.org/https://doi.org/10.48550/arXiv.1506.02640

Sinaga, A. S. R. (2017). Implementasi Teknik Threshoding Pada Segmentasi Citra Digital. Jurnal Mantik Penusa, 1(2), 48–51.

Suartika E. P, I Wayan, Wijaya Arya Yudhi, S. R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101. Jurnal Teknik ITS, 5(1), 76. http://repository.its.ac.id/48842/

Wicaksono, B. A., Yuniar Purbasari, I., & Vita Via, Y. (2021). Deteksi Objek Mobil dan Motor pada Lalu Lintas Berbasis Deep Learning. Jurnal Informatika dan Sistem Informasi, 2(2), 334–342. https://doi.org/10.33005/jifosi.v2i2.284

Zulkhaidi, T. C. A.-S., Maria, E., & Yulianto, Y. (2020). Pengenalan Pola Bentuk Wajah dengan OpenCV. Jurnal Rekayasa Teknologi Informasi (JURTI), 3(2), 181. https://doi.org/10.30872/jurti.v3i2.4033