Implementation of Convolutional Neural Network Method for Highway Damage Classification

Dharma Aryani[a], Syahdilla[b,*], Syahrir[c]

[a,b,c] *Electrical Engineering Study Program, State Polytechic of Ujung Pandang*

[*]*Email: dharma.aryani@poliupg.ac.id*

***Abstract*— Road damage can be divided into several types including cracks, distortion, wear, fattening, and holes. This research aims to classify the type of road damage using Convolutional Neural Network (CNN) on *1* algorithm. The dataset used for the training process is carried out using a dataset obtained from the roboflow platform which amounts to 2,200 data. While the dataset used for the testing process uses data in the form of digital images as much as 20 real data with damaged road conditions captured using drones and 10 data captured via satellite. The model used to classify road damage is a model obtained from training results that has the highest f1-Score and mAP@0.5 values and less response time, namely training conducted using the 20th epoch parameter. In this study, system testing was carried out using three test scenarios, namely tests carried out using images captured from two different locations and had no noise in them, then tests carried out using images that had noise such as transverse network cables or passing vehicles, and tests carried out using images captured via satellite. Based on the test results that have been carried out, it can be concluded that the Convolutional neural Network (CNN) method on the *yolov5s* algorithm can be used in classifying road damage with very good accuracy. Of the three test scenarios carried out, the first scenario obtained an accuracy of 95%, the second scenario only obtained an accuracy of 87%, and the third scenario obtained an accuracy of 98%.**

***Keywords*— Road Damage; Convolutional Neural Network; Yolov5s.**

I. Introduction

Roads are land transportation facilities that play an important role in the field of transportation that connects one city with another city, between cities and villages, and between villages and other villages [1]. The increasing intensity of vehicles that are increasing every year and along with the weather conditions in Indonesia, the burden that must be borne by the highway is increasing every day [2] as a result, many roads are damaged such as cracks and potholes, which can increase the risk of traffic accidents. Traffic accidents that occur are generally attributed to the negligence of motorists[3] but in fact, due to the large amount of damage on the road, it is difficult for drivers to find out where the damage is located on the highway, which triggers accidents [4]. Detection of road damage so far is still using manual methods, namely by direct observation with the sense of sight.

This manual method is considered less effective because it takes a long time, and is dangerous due to the high intensity of vehicles passing by, as well as subjectivity and fatigue factors [5]. Based on the above factors, a study is needed to find a new method that has a high level of accuracy in identifying the level of damage to the highway.

In some previous studies, road damage classification has been carried out, including research that classifies the level of road damage using Image Processing methods with Thresholding and K-Means Clustering algorithms [6], and based on the results obtained in the study, the Thresholding and K-Means Clustering algorithms can identify the level of road damage well and effectively because the algorithm is able to identify road damage in 21 images from the total test data of 27 images. In addition, there is also research that identifies road damage using the Edge detection algorithm, and the accuracy obtained from the research is 80% because the system can identify 8 out of 10 total data tested [7]. However, in previous studies, classification based on the type of damage that occurs on the highway has not been carried out because it only identifies the location of damage in the image and the level of damage in the image so that the type of damage contained in the image is not known with certainty.

Therefore, a system is proposed that can assist related parties in obtaining information about the condition and Janis damage that occurs on the highway, namely by using the Convolutional Neural Network method by utilizing the yolov5s (You only

Look Once) algorithm to classify damage to the highway. The Convolutional Neural Network method is one part of the deep learning method that has good accuracy results [8], and YOLO is one of the algorithms in CNN that has a fast identification process because it makes object detection a regression problem [9].

II. Materials and Method

The design and manufacture of a highway damage classification system is carried out in several stages, including literature study, system design, system creation which includes selecting a pre-trained weight model for training data, labelling data using the yolov5s pre-trained weight model, and testing system performance based on the optimization value obtained from the test results.

A. Literature Study

The initial survey carried out is by identifying problems that aim to find out the causes and impacts of damage to the highway. Then the Literature Study is carried out by exploring the material and collecting analyzes related to the methods used in system design.

B. System Design

1) Overview of System Creation

The data used in this study is data in the form of images with damaged asphalt conditions such as cracks, holes or both. The data division process where at this stage the data is divided into two categories, namely Training data and Test Data. Furthermore, the data annotation stage (Labelling) which is a process where the data is labeled in the form of a bounding box around the object. Then proceed to the system design stage, this system design utilizes one of the methods in the yolov5s algorithm, namely the convolutional neural network method.

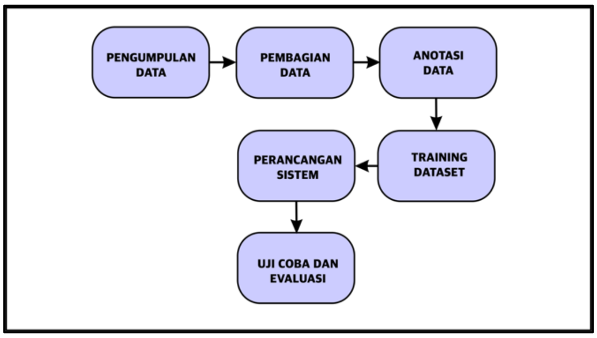

Fig. 1. System Creation Flow

2) System Block Diagram

The block diagram of the system creation process that explains all the stages in the working process of the Highway Damage Classification system is shown in Fig. 2.

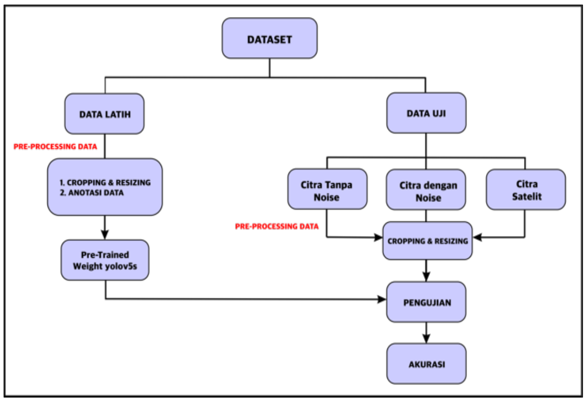

Fig. 2. System Block Diagram

The initial process of making the system is by collecting datasets in the form of digital images obtained from the results of capture with a drone camera, then the pre-processing stage which aims to improve image quality and equalize the size of each image. Then the program for the system is made using Visual Studio code software using the python programming language. Data that has gone through the pre-processing stage will be processed using the Convolutional Neural Network method in the yolov5s algorithm so that it will produce an output in the form of an image of the classification results of highway damage.

3) Use-Case Diagram

Use-Case Diagram is a description of the interaction between the User and the system. In addition, the Use-Case Diagram also serves to display the parts, functions and features contained in the system as shown in Figure 3.

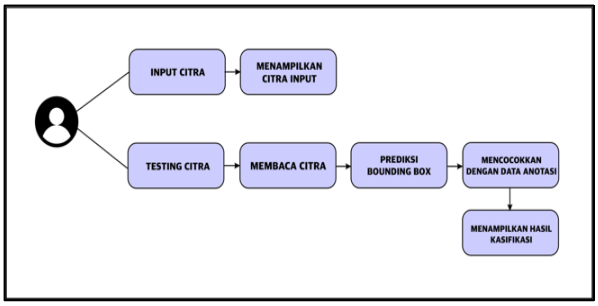

Fig. 3. Use-Case Diagram

Fig. 3 shows the features provided to users in the Road Damage Classification system. Initially, the user will run the program and input data in the form of digital images into the system, then the system will display the selected image into the system, then the user will test the data, the initial stage that will be carried out when testing the data is to predict the bounding box and match the labeling data using the training model that has been imported into the program, when the data matches and matches the class of data that has been labeled at the data annotation stage, the classification results of the image will appear marked with a bounding box on each object.

4) Block Diagram of Work Process on the User Side and on the System Side

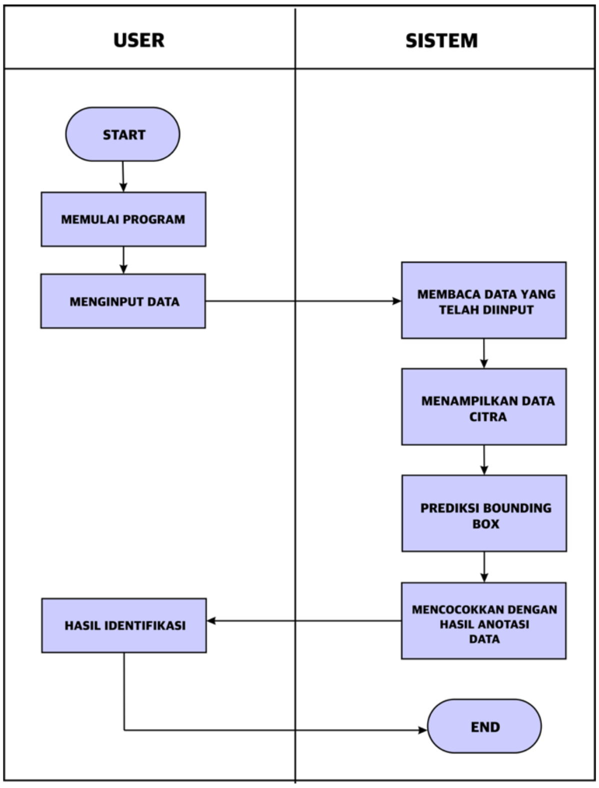

Fig. 4. User and System work Process Diagram

Fig. 4 is the Work Process on the User and System side. This block diagram describes the flow of activities contained in the system. In the block diagram above, it is illustrated that after the user runs the program on the system, then the user will input data in the form of images into the system. After the user inputs the data, the system will process the data. The first stage carried out by the system in processing the input data is by reading the input data first, then the data that has been read will be displayed by the system then the data will be matched with the results of data annotation which is data that has been labeled with Bounding Box. Data that matches and matches the annotated data is then classified based on its respective class so that in the final stage it will display the classified image.

C. System Building

The Highway Damage Classification System was created using Visual Studio Code software. The following are the stages of the process of making a Highway Damage Classification System.

1) Pre-Trained Weight yolov5s

To perform training using the yolov5s algorithm, a pre-trained weight model of yolov5 is required which is downloaded from the github platform.

2) Data annotation

The data that will be trained is data that comes from the roboflow platform which amounts to 2,200 images, each class consisting of 550 images. The dataset is an image that has been preprocessed by cropping each image so that the image is more focused on the main object in the image, which in this case is the condition of the damaged highway, then the image is resized with a size of 640x640 pixels to uniform the size of each image which was previously different. The data is data in the form of damaged highway conditions that have been labeled using 4 classes of damage to the highway, namely Alligator Crack, Longitudinal Crack, Pathole, and Transverse Crack.

In roboflow the data labeling process is carried out by giving a bounding box to each object contained in the image and adjusting based on the class of each object such as longitudinal crack, transverse crack, pathole or alligator crack, then the dataset that has gone through the pre-processing stage and the labelling stage and then split into three folders, namely the train folder containing images that will go through the training stage, the valid folder containing images that will be validated from the model prediction results and the test folder containing images that will be tested using the model from the training results. In addition to the folder containing image data, there is also a file in .yaml format which is a file containing the class names of objects that have been labeled which then the .yaml file will be called into the program during the training process to adjust the objects in the training data with the class from the labelling results.

3) Data Training

Dataset training is done using the yolov5s (You Only Look Once) algorithm. The data that is trained is the one that has gone through the labeling stage. However, the main thing that needs to be prepared before the training process is the pre-trained weight model that will be used for training. The model used is a model from yolov5s obtained through the github platform.

The data training process is carried out using different training parameters so that the data training is carried out several times. In this study, the data was trained 5 times, each training using a different number of epochs including 5, 10, 15, 20 and 100 epochs. Training data using a different number of epochs is done to see a comparison of the prediction accuracy of the models generated from each epoch that has been trained. Then, data training is again carried out 4 times using epochs that produce the best model carried out with different batch values to compare the speed of the algorithm in training. The batch values used are 20, 50, 65, and 75.

4) System Testing

System testing was carried out by testing 20 datasets captured by drones and 10 datasets captured via satellite using the model obtained from the training results using the Yolov5s algorithm. In this study, system testing was carried out using three test scenarios, including:

1. Testing carried out using images captured from two different locations and have no noise in them.

2. Testing carried out using images that have noise such as network cables that cross the highway, and passing vehicles.
3. Testing conducted using images captured via satellite.

The success of the model in classifying objects is measured by the optimization value resulting from tests carried out with these three scenarios, where the higher the value produced, the better the performance of the model in making predictions.

III. Result and Discussion

This chapter will discuss the test results of the method used in the road damage classification system, namely the Convolutional Neural Network method on the yolov5s (You Only Look Once) algorithm. In this research, system testing was carried out using a dataset captured using a drone.

1) *Training Dataset Result*

The dataset is trained using different training parameters, namely training carried out using a different number of epochs and training carried out using different batch values. Training data with different parameters is done to compare the value of the training results used to determine the best model in making predictions. In addition, to complete the model validation process, the confusion matrix method is used to obtain the model with the best performance to classify highway damage.

The value of the training results used to determine the best model in making predictions is seen based on the f1-score value and mAP@0.5. Mean Average Precision at 0.5 (mAP@0.5) is a metric used to evaluate the performance of the model in detecting objects. Although the performance measurement of the model can also be seen from mAP@0.5, the two have differences because mAP@0.5 serves to measure the accuracy of the model in finding objects, while f1-Score is used to measure the performance of the model in finding all objects contained in an image.

Fig. 6 is the f1-score curve obtained from the training results using the 20th epoch. The figure is a curve that shows the f1-score value of each class and the entire class from the training results, so from the figure it can be seen that the f1-score obtained for the entire class is 79%, the alligator crack class obtained an f1-score of 89%, the longitudinal crack class obtained an f1-score of 71%, the pathole class obtained an f1-score of 78%, and the transverse crack class obtained an f1-score of 82% which if these values are close to 1. 0 or 100 then the model has the best performance in finding all objects in an image and has a good balance in making predictions and conversely if the f1-score is 0, the value means the worst performance of the model in detecting all objects in an image.

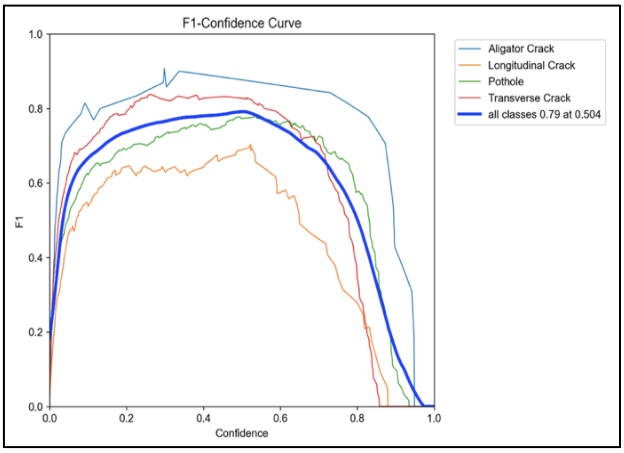

Fig. 5. F1-Score 20th Epoch

Fig. 4. is the f1-score curve obtained from the training results using the 20th epoch. The figure is a curve that shows the f1-score value of each class and the entire class from the training results, so from the figure it can be seen that the f1-score obtained for the entire class is 79%, the alligator crack class obtained an f1-score of 89%, the longitudinal crack class obtained an f1-score of 71%, the pathole class obtained an f1-score of 78%, and the transverse crack class obtained an f1-score of 82% which if these values are close to 1. 0 or 100 then the model has the best performance in finding all objects in an image and has a good balance in making predictions and conversely if the f1-score is 0, the value means the worst performance of the model in detecting all objects in an image.

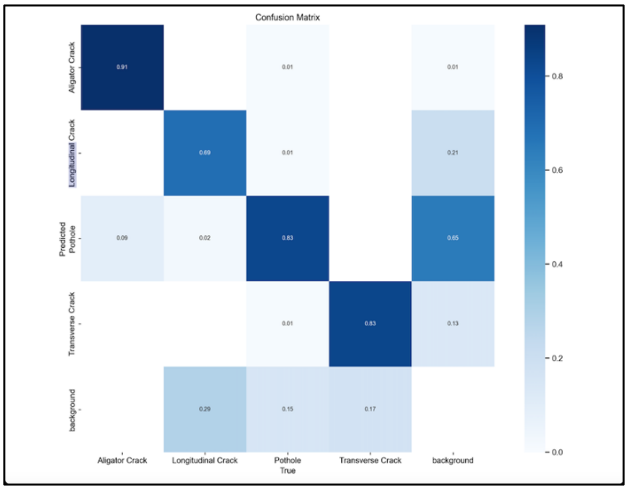

Fig. 6. Confusion Matrix 20th Epoch

Fig. 7 shows that the model obtained from the training results with the 20th epoch has been able to predict the Alligator Crack class by 91%, Longitudinal Crack class by 69%, Pathole class by 83%, and Transverse Crack class by 83%, this shows that the higher the True Positive (TP) value of the object prediction results, the better the performance of the model in predicting.

Table 1. F1-Score and mAP@0.5 value of different epochs

No.	*Epoch*	*F1-Score*	*mAP@0.5*
1	5th *Epoch*	0.71	0.75
2	10th *Epoch*	0.80	0.81
3	15th *Epoch*	0.80	0.82
4	20th *Epoch*	0.79	0.83
5	100th *Epoch*	0.81	0.83

Table 1 shows the F1-Score and mAP@0.5 values of all classes from training results using different number of epochs, namely the 5th, 10th, 15th, 20th and 100th epochs. In the table, it can be seen that from the overall value obtained from training conducted using a different number of epochs, it can be seen that training using the 20th and 100th epochs obtained high f1-score and mAP@0.5 values compared to other epochs.

Training conducted using the 20th epoch produces the same mAP@0.5 value as mAP@0.5 from the training results with the 100th epoch, which is 0.83%. The f1-score value generated from both does not have a much different difference. However, even though the mAP value of both is the same, the response time of the 20th epoch and the 100th epoch when training is certainly different because the 20th epoch has less response time than the 100th epoch.

Table 2. Training Results using different Batch Value

Epoch	*Batch*	*F1-Score*	*mAP@0.5*	*Time*
20	20	0.79	0.83	5.93 *Hours*
	50	0.79	0.83	5.79 *Hours*
	64	0.79	0.83	5.66 *Hours*
	75	0.79	0.83	5.57 *Hours*

Table 2 is the result of training data performed using the same epoch parameters, namely the 20th epoch but with different batch values. The table shows that training done using the 20th epoch with a batch value of 20 requires a training time of 5.93 hours, while training done using the 20th epoch with a batch value of 75 only requires a training time of 5.57 hours. Based on these results, it can be seen that the number of batches will affect the training time, because the larger the batch value used, the more data is processed in an iteration, thus speeding up the training process. In addition, changing the batch value during the training process does not affect the F1-Score and mAP@0.5 values of the training results because the batch only affects the time required to perform the training process.

2) *Testing System Result*

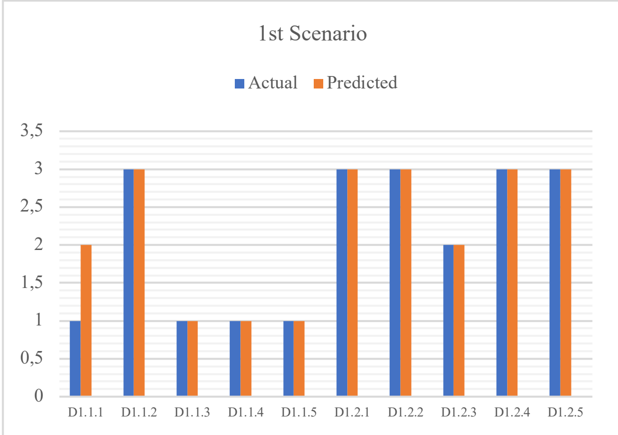

Fig. 7. First Scenario Testing

Fig. 9 is the result of testing conducted in the first scenario using real data in the form of images that have no noise. In the table, it can be seen that of the 10 number of images tested, there is 1 image that is not detected properly, namely data1.1.1.jpg. In the image, it can be seen that the model predicts two objects, but of the two objects successfully predicted by the model, not all of the prediction results are correct because there is one object that is not correctly predicted by the model. The object that should not be predicted by the model is tire friction located around the object that should be predicted, but the model predicts the tire friction as one type of road damage, namely the transverse crack class because the object almost resembles one type of road damage. From the tests that have been carried out in the first scenario using images in the form of images that do not have noise, 95% accuracy is obtained because of the 29 objects that have been successfully predicted by the model, only one object is not predicted correctly so that based on the accuracy value obtained it can be concluded that the model is able to predict images in the form of highway damage conditions that do not have noise or interference in them with good accuracy.

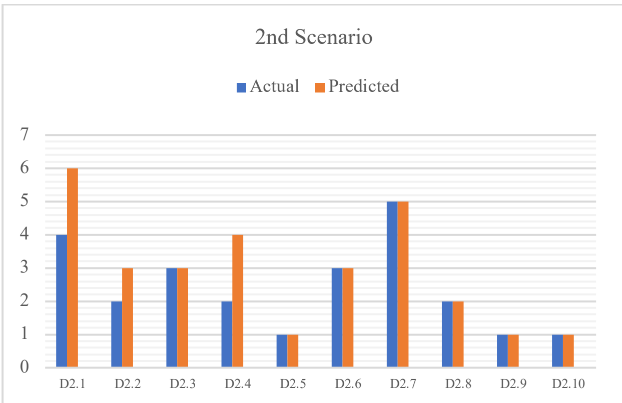

Fig. 8 Second Scenario Testing

Figure 10 is the result of testing conducted in the second scenario. The test is carried out using images that have noise around objects that should be detected such as network cables, or passing vehicles. In the table it can be seen that of the 10 images tested there are 3 images that are not predicted properly by the model, namely data image 2.1.jpg, data 2.2.jpg and 2.4.jpg. In data 2.1.jpg it can be seen that the model predicts 3 objects in the image as transverse crack, but the model should only predict 1 transverse crack so that of the 3 predicted transverse cracks, two of them are not predicted correctly because the model predicts a network cable that crosses the highway as one type of damage, namely transverse crack because the cable almost resembles one type of damage to the highway. In image 2.2 there are 2 types of objects that should be predicted, namely longitudinal crack and transverse crack, but in the image it can be seen that there are 3 types of damage that are predicted by the model because the model predicts one of the passing vehicles as one type of highway damage, namely pathole, the same thing also happens in the test conducted on data2.4.jpg. In the picture, it can be seen that the model is able to predict as many as 4 objects in the image, but 2 of them are not predicted properly because the objects that should be predicted by the model are longitudinal crack and transverse crack only, but the model predicts 2 motorized vehicles as one type of damage, namely pathole so that there are 4 number of objects predicted by the model.

Although the tests carried out on each data do not produce perfect predictions, the tests in this second scenario do not produce low average accuracy, but on the contrary, they get an accuracy of 87% so it can be concluded that the model is able to predict images that have noise with fairly good accuracy because of the 29 number of objects predicted, only 5 objects are not predicted correctly.So it can be concluded that the yolov5s algorithm is not only able to predict images that do not have noise but also images that have noise.

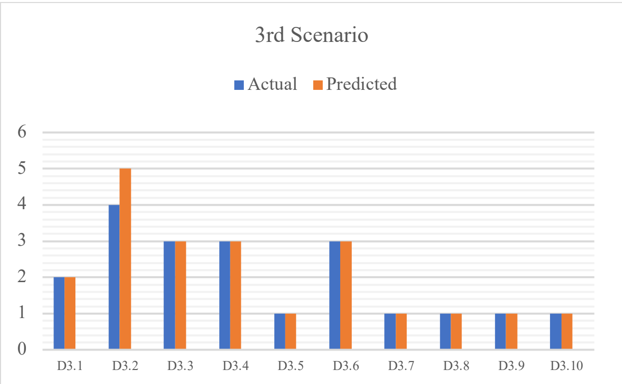

Fig. 9 Third Scenario Testing

The results of the tests carried out in scenario 3 are shown in Figure 11. In this test, of the 10 data tested in scenario 3, only 1 data was not classified properly, namely data 3.2.jpg. In the image, it can be seen that there are only 4 objects that should be predicted by the model, namely two objects with the transverse crack class and two objects with the longitudinal crack class, but the model predicts a motorcycle that is passing by as one type of damage, namely a pathole. Nevertheless, from all the data tested in scenario 3, an accuracy of 98% was obtained, which shows that the model is also able to predict the type of road damage on satellite-captured images.

IV. Conclusion

This research was conducted to classify the type of damage on the highway using Convolutional Neural Network (CNN) on the yolov5s algorithm. In this study, testing was carried out using three test scenarios, including tests carried out using datasets captured from two different locations without having noise, and tests carried out using datasets that have noise and datasets captured via satellite. There were 10 datasets tested for each test scenario. The model used to classify highway damage is a model obtained from training results that has the highest f1-Score value, mAP@0.5 and less response time, namely training conducted using the 20th epoch. From the results of tests that have been carried out using two scenarios, an accuracy of 95% was obtained for testing the first scenario, an accuracy of 87% obtained from tests carried out using the second scenario, namely with datasets that have noise, and an accuracy of 98% for tests carried out using the third scenario, namely datasets captured via satellite. Based on these accuracies, it can be concluded that the Convolutional Neural Network (CNN) method in the yolov5s algorithm can classify road damage with very good accuracy.

Acknowledgment

We would like to thank our institution for providing the facilities and support needed to carry out this research, and we would also like to thank our colleagues who provided valuable insights and inputs in developing the ideas and methodology in this paper. Your contributions have been invaluable to the success of this research.

References

[1] Iqbal Hastri Firmandani and Agus Trilaksana, “Perkembangan Transportasi Darat di Probolinggo (BUS),” *Journal Pendidikan Sejarah*, vol. 11, no. 3, 2021.

[2] N. Fauziah and A. Verani Rouly Sihombing, “Prosiding The 13th Industrial Research Workshop and National Seminar Bandung,” 2022.

[3] T. Tjahjono and D. Maulina, “Kelalaian Manusia (Human Error) Dalam Kecelakaan Lalu Lintas: Analisis Berdasarkan Pemrosesan Informasi Human Error in Traffic Accidents: Analysis of Based on Information Processing,” 2018.

[4] A. E. Wijaya, W. Swastika, and O. H. Kelana, “Implementasi Transfer Learning Pada Convolutional Neural Network Untuk Diagnosis Covid-19 Dan Pneumonia Pada Citra X-Ray,” 2021.

[5] A. N. Utomo and N. Lestari, “Aplikasi Deteksi Kerusakan Jalan Raya Menggunakan Algoritma K-Nn (K-Nearest Neighbour) Road Detection Application Using K-Nn Algorithm (K-Nearest NeighbouR),” vol. 10, no. 1, 2021.

[6] Nurfiyah, Yan. Rianto, and Dwiza. Riana, “Identifikasi Tingkat Kerusakan Jalan Raya,” vol. 13, no. 1, pp. 34–44, 2021.

[7] A. Suryowinoto, A. Hamid, I. T. Adhi, and T. Surabaya, Penggunaan Pengolahan Citra Digital Dengan Algoritma Edge Detection Dalam Mengidentifikasi Kerusakan Kontur Jalan. 2017.

[8] T. Bariyah and M. Arif Rasyidi, “Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada

Motif Batik Convolutional Neural Network for Multi-Label Batik Pattern Classification Method," 2021.
[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection."[Online].Available:http://pjreddie.com/yolo/