# Road Shoulder Classification Using The CNN Algorithm with The MobileNetV2 Model

Rahardhita Sudibyo,And Haniah Mahmudah

Departement of Electrical Engineering
Politeknik Elektronika Negeri Surabaya
Surabaya, Jawa Timur, Indonesia

widi@pens.ac.id, haniah@pens.ac.id .

**Abstract.** The condition of the road that is traversed by high and repeated traffic volumes will affect the condition of road construction, leading to a decline in the quality of the road, which impacts the safety, comfort, and smoothness of traffic. This paper will discuss Deep learning to evaluate damage detection models and classify large- scale road shoulder surface data sets in advanced Convolutional Neural Network (CNN) algorithms. One of the models that maintain high accuracy and produce better results is Mobilenet V2. From the MobileNet V2 model, it has been confirmed to obtain the best accuracy results for each epoch and batch size for shoulder parameter classification are an accuracy rate of approximately 0.7 to 0.8 with a minimum loss value of 0.2 to 0.3; thus, using the MobileNet V2 model to classify the shoulder yields the optimal results.

**Keywords:** CNN, MobileNet V2, Road Shoulder Surface

## 1. Introduction

The road is an essential means of land transportation in the community by providing convenience and facilitating economic and cultural activities between regions in Indonesia [1]. Various community activities are directly proportional to the traffic load that affects road conditions; the more significant the community activity, the greater the traffic load [2]. Low road quality will negatively impact road users by increasing fuel consumption and vehicle maintenance costs, reducing driving comfort, and possibly jeopardizing traffic safety [3]. Furthermore, the process of checking and observing the road surface condition is entirely manual, resulting in inefficiency and the requirement for long hours to inspect road conditions visually [4].

According to data from the East Java Highways Public Works Department, overloaded vehicles caused many damaged roads and accidents in the last three years, from 2017 to 2020 [5]. Meanwhile, cracks in the concrete pavement of the road shoulder in the Cageur - Ciamis Road Section was caused by the concrete's rigid nature. Moreover, no dowel to withstand the deflection and tensile loads encountered during this work's execution [6].

Currently, the number of vehicle users in Indonesia is rapidly increasing, and unexpected emergencies such as burst tires, engine damage, and other occurrences while driving. Due to the emergency, adequate road infrastructure, specifically the road shoulder, is urgently required [7]. National and provincial highways are heavily


M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_24

travelled, and numerous still have unpaved shoulders since the unpaved shoulder are susceptible to scouring. It will result in numerous losses, including the significant elevation difference between the road body and the road shoulder. The scour on the road's shoulder will become more extensive, more prolonged, and more profound over time. If this occurs, the foundation layer beneath the roadway cannot hold the weight of passing vehicles. Consequently, the road has been damaged. Although the strength of the road body is strongly supported by the road shoulder [8].

The SSD detection model consistently produces a low mean mAP value (less than 20 per cent) because all detection models are based on CNN; small damage loss information may be lost during the CNN network during image size reduction to extract features. The disadvantages of these models are minor, particularly when detecting road damage since the built-in camera can approach the damaged object as the vehicle moves forward. In this instance, the damage is minor and can be overlooked since it does not affect the road's operability. For each type of damage, all models use the parameter IoU 0.5. With AP values of 71.64 per cent, 45.28 per cent, and 70.77 per cent, respectively, the Faster R-CNN Inception Resnet V2 model was the most efficient. The Inception V2 SSD achieves performance comparable to the Faster R CNN Resnet50 by producing an AP value of 11.97 per cent. Faster R CNN produces good recall values for all IoU parameters, with Inception V2 achieving the highest average values for medium objects, large objects, and overall of 41.10 per cent, 58.05 per cent, and 53.06 per cent, respectively. The results indicate that Inception V2 outperforms its peer feature extraction network. SSD Inception V2 outperforms other SSD detection models, with a maximum AR value of 41.44 per cent for small object detection and a maximum AR value of 46.61 per cent for significant object detection [10].

Based on the problems and previous research, this research employed deep learning to develop a large-scale road damage data set in the advanced Convolutional Neural Network (CNN) algorithm, one of the models that maintain high accuracy and generates better results, namely MobileNet V2. MobileNet V2 utilized lightweight depth-wise convolutions to filter features in the expansion layer used for feature extraction [11]. MobileNet V2 as a pre-trained model utilizing the TensorFlow Object Detection API could generate great accuracy and detectable area for the presence of each category of objects in an image. The efficiency in object detection was about 85.18% above the average. The loss per step or epoch was 2.73 (under 3), which oversees the model's reliability [11].

This paper presented a shoulder classification optimization using a deep learning convolutional neural network with the MobileNet V2 model. This research began as a project enhancement focused on machine learning and real-world implementation, as referenced in previous works. [12][13]. The optimization process was carried out by adjusting the hyperparameters and comparing the results with changes in each epoch and batch size to find the best results with this lightweight and fast model. From the MobileNet V2 model, an accuracy rate of about 0.7 to 0.8 with a minimum loss value of 0.2 to 0.3 was obtained; thus, this paper employed the MobileNet model to classify the shoulder to produce optimal results.

## 1. RELATED WORKS

The authors discuss related works to this paper briefly in this section. The YOLO method, which employed a Convolutional Neural Network (CNN) in its architecture, has generated promising results in object detection on both images and videos. YOLO has been tested on various datasets and produced faster and more accurate results. Cropping and resizing images, followed by annotating the data, are the pre-processing steps in this study. Subsequently, the training was completed by fine-tuning the YOLO network process. The YOLO architecture employed nine convolution layers and six max pool layers. The dataset testing results demonstrated that the highest accuracy was 99 per cent and the highest average IoU is 75,1%. The classification run time was 0.883 seconds per image [14].

Convolutional neural network to train a crash detection model with data sets and compare the accuracy and runtime speed on both, using a GPU server and a mobile device, a convolutional neural network will train a crash detection model using data sets and compare the accuracy and runtime speed by annotating boundary boxes representing the location and type of damage. Regarding the specifics of the dataset, the models utilized by this system were SSD using Inception V2 and SSD using MobileNet. In this sense, the Inception V2 SSD is twice as slow as the MobileNet SSD. Consequently, the precision of the detection results was more than 75%.

Road damage detection using a state-of-the-art object detection model with a simple and efficient model with a high-level semantic representation. Faster-RCNN and SSD are examples of art. For training and testing, Faster-RCNN used open-source code based on MXNet. They are using open-source code based on Caffe while on the SSD. SSDs typically outperform Faster-RCNN, particularly when it comes to detecting small objects. If computing resources are limited, SSDs can detect larger images than Faster-RCNN, so results can be improved by feeding larger images. For the same input size, Faster-RCNN performs slightly better than SSD [15].

## 2. Road Shoulder Surface Classification Using CNN

### 2.1 Machine Learning Model Training

In this study, the MobileNetV2 model was employed for image classification, and the model's probability was emphasized. MobileNetV2's primary structure utilized Depthwise Separable Convolutions (DSC) techniques for portability and not only improved the information destruction in non-linear layers in convolution blocks by employing Linear Bottlenecks but also introduced a new structure called Inverted residuals to preserve the information [16].

The contracting path adheres to the typical architecture of a convolutional network by employing two convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2 $x$ 2 max pooling operation with stride 2 for downsampling. These three processing stages, referred to as a "block," are repeated multiple times (creating a deep network), resulting in a set of fully connected layers (classifier stage).

The convolution layers obtain locally weighted sums (termed "feature maps") at each layer by computing filters that are repeatedly applied to the complete data set to enhance training efficiency. Consequently, the non-linear layers enhance the non-linear qualities of the feature maps. The pooling layer subsamples non-overlapping regions in feature maps, enabling the network to aggregate local features and discover complex features. At each downsampling stage, the number of feature channels is doubled. Max pooling selects the most significant element from the corrected feature map and selecting the most significant element may also imply selecting the average pooling [17].

At the end of the MobileNet V2 stage, the feature map matrix was flattened into vector form and fed into the classifier stage, a fully connected layer similar to a neural network. A model was created by combining these features with fully connected layers.

The API provides test images for image object detection. Subsequently, these images are directed to the tensor flow serving server. TensorFlow Serving is a flexible, high-performance serving system designed for use in production environments with various machine learning models. TensorFlow Serving simplifies the deployment of new techniques, algorithms, and experiments while retaining the similar server design and APIs [18].

## 2.2 Dataset Description and Environment Setup

This dataset was obtained by direct data collection, the Public Works Service, and the internet by visiting the kaggle.com website, which subsequently downloaded numerous images, bringing the total number of street images obtained to 10,350. As depicted in Figure 1's overview of the dataset used, each photograph has a unique file size, format, and pixel size and has been individually annotated. The entire dataset was divided into 80% as training data and 20% as testing data.

(a). (b).

**Fig. 1.** Overview of Dataset: (a) Road Shoulder; (b) Not Road Shoulder

The dataset is pre-processed to generalize the entire image's size and format to speed up the performance, improve model performance, shorten the training process, and vary the amount of RAM used. Thus, using the Windows instrument, the image data that has been collected is converted into an image with a resolution of 640 *x* 480 pixels. The training process is carried out with experimental equipment and software demonstrated in Table I.

**TABLE I.** ENVIRONMENT SETUP

| **Device Name** | **Version** |
|---|---|
| Operating System | CentOS Stream |
| CPU | Ryzen 9 5950x |
| GPU | RTX3080 10 GB |
| RAM Server | 32 GB |
| Tensorflow | 2.0 |
| Python | 3.8.3 |

### 2.3 Scenario CNN Modeling for Model MobileNet V2

Annotating datasets categories road shoulder surface datasets obtained from the overall dataset collection. The road shoulder surface was divided into two categories: datasets that include the shoulder and datasets that do not include the shoulder. Subsequently, the dataset was uploaded to Python for processing using a Convolutional Neural Network (CNN) and the MobileNet V2 model. Fig. 2 illustrates the proses of implementing the CNN Model.

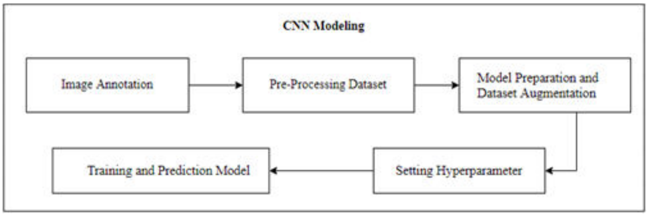


**Fig. 2.** The Process of Implementing the CNN Model.

Typically, the Pre-Processing procedure is included into the training process. In addition, preparation is required before carrying out the dataset training process, namely a virtual environment to facilitate the training process, where this environment is in the form of files prepared in the training process. The training file is divided into two datasets, namely training and validation datasets, which were separated randomly and automatically in a library. Following the partitioning of the datasets is the augmentation process.

In the Convolution Neural Network, the more datasets are trained, can reduce the occurrence of overfitting, and get better the model performance. One method of expanding the dataset is image augmentation, a technique of manipulating an image without losing its essence or essence. In this instance, augmentation worked in the process of playing and mirroring data on each image; thus, the amount of data obtained will increase and vary.

This stage was a stage in making a model using hyperparameter settings. The model used was base_model (MobileNet v2), where the model preparation process was only inputted without having to write down the existing layers since the model has already been entered into the library. Subsequently, the existing model was added to other models in front of it; thus, it could receive the input. The modelling in this training process was global_avarage_layer and dropout_layer, and the output used the predictor layer model. Subsequently, from modelling, it was necessary to enter training parameters into variables for the training process to be carried out. The model performance results were affected by the hyperparameter tuning used; therefore, it was essential to set and determine the correct parameters in the training process.

In the training process, parameters that can be changed to produce good accuracy are found at a given epoch and batch size. Model experiments on various parameters of batch size and epoch size are carried out as part of the model optimization process to find good performance. Batch size determines how many samples of the dataset are taken at each stage of the training process so that the samples represent the entire dataset for each weight change in the neural network. While the number of epochs is a hyperparameter that determines how many times the learning algorithm work to process the entire training dataset. One epoch means that each sample in the training dataset has the opportunity to update the internal model parameters. This weight change is based on how much loss the model has, which determines the quality of the model. The overall testing scenario for model MobileNetV2 is represented in Table II.

**TABLE II.** HYPERPARAMETER SCENARIOS

| **Hyperparameter** | **Testing Scenarios** |
|---|---|
| Batch Size | 16, 24, 32, 100 |
| Epochs | 10,20,30,40,50,60,70,80,90,100 |
| Optimizer | RMSprop |
| Learning Rate | 0.0001 |

After that, a prediction is made on the test dataset to see if the dataset is in accordance with what is desired based on the declaration given, namely the shoulder of the road and not the shoulder of the road. Thus, it can be seen from the prediction results that the image predicted by machine learning is appropriate.

## 3. Results And Analysis

This test was carried out as a result of research testing with the accuracy of the MobileNet V2 model using the Confusion Matrix, which is a matrix that describes the comparison of the classification results carried out by the model with the actual classification results so that this test can classify the shoulder of the road and not the shoulder of the road using the best model and its accuracy is calculated using confusion matrix as shown in (1). In this equation test to see the level of loss and accuracy, changes are made to the epoch and batch size.as in:

$$Accuracy = \frac{\mathrm{TP+TN}}{TP+TN+FP+FN} \quad (1)$$

The *TP* is True Positive, *TN* is True Negative, *FP* is False Positive, and *FN* is False Negative

### 3.1 MobileNet V2 with Epoch changes

Based on the hyperparameters in Table 2, the results of the best loss training and validation losses used were batch size 32 and epoch starting from 10 to 100. The training results at 10 to 40 epochs resulted in the level of training loss, and validation loss did not have a significant loss difference, as illustrated in Fig 3. The graph results that occurred in epoch 10. In epoch 10 the training loss was 0.3594, and the validation loss was 0.3814. Thus, the difference between the two losses was minimal, which was 0.0.

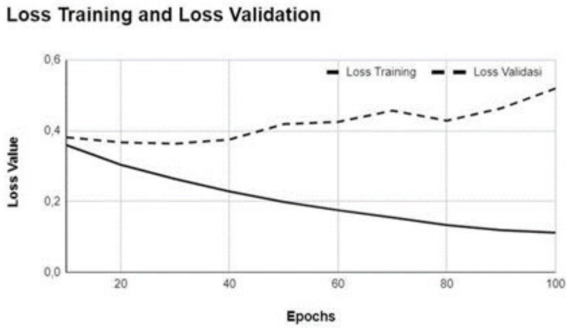


**Fig. 3.** Loss training and validation with 10 until 100 epochs using MobileNet V2.

.

The results of the training dataset in epoch 10 were not overfitting. The overfitting condition occurred when the model memories all the training data without analyzing the validation of the data since the model was overly focused on a particular training dataset; consequently, it could not generate accurate predictions for validation if other similar datasets are provided.

The accuracy of the classification road shoulder system developed is increasing as the number of epochs increases. However, there was a slight fluctuation in loss, specifically at epoch 50 until 100; the system's loss performance increased dramatically, only to drop again in the following additional epoch, as demonstrated in Fig 3.

Theoretically, the system's performance improves as the epoch value increases since the system can better generalize data based on updates to weights and biases or lessons learned in the previous epoch. Such value fluctuations result from the system's inability to accurately generalize or anticipate the data due to overfitting.

### 3.2 MobileNet V2 with Batch_size changes

Based on MobileNet V2 model the hyperparameters in the Table 2, results of the best loss training and validation losses used batch size change. The comparison batch sizes of 10,16, and 100. Although the numbers compared were small, they were representative because we could view the effect of using different batch sizes. The training results in batch sizes 10,16 and 100 with 10 epochs resulted in a training loss level that did not have a significant loss difference as depicted in Fig. 4, 5, and 6 of the training loss graphs.


**Fig. 4.** Training Loss MobileNet V2 model batch size10.

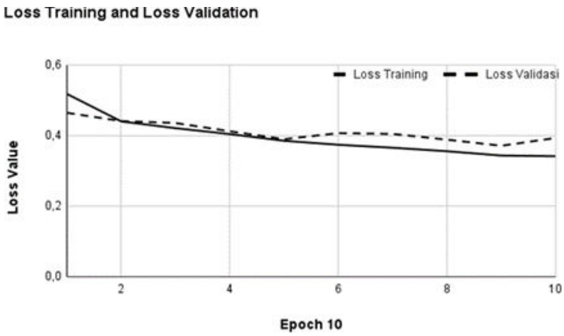


**Fig. 5.** Training Loss MobileNet V2 model batch size16.


**Fig. 6**. Training Loss MobileNet V2 model batch size 100.

Figures 4, 5, and 6 demonstrate that the larger the batch size used, the higher the loss number obtained. A smaller batch size leads to a lower loss number. Typically, large batch sizes are utilized since they permit the possibility of computing acceleration. Using a tiny batch size will need a great deal of time. However, computing acceleration comes with a cost associated. A batch size that is highly large will produce less- than-ideal results. The larger the batch size, the less thoroughly the results are examined. Typically, users of deep learning will evaluate the tools' capabilities, the time necessary for a lengthy training procedure, and the potential for yield optimization.

### 3.3 Prediction for Shoulder Parameters

Based on the training results are demonstrated in Table 2 as the utilized hyperparameter, predictions were made from the dataset where the training dataset was tested from ground truth or original data road shoulder that the test results of the evaluated dataset have an accuracy of 0.7812 and a loss of 0.3902. The MobileNet V2 model obtained is relatively good since when the dataset test was carried out, the loss results obtained were smaller with higher accuracy. Therefore, the results of the prediction image dataset test were obtained after the training process employed 10 epochs with an error rate between label and prediction of 6.25%, which was manually calculated road shoulder to determine whether the test data generated was appropriate or not, as depicted in Fig 7. where prediction was the road shoulder image of the training results. The label represents the ground truth or the original road shoulder image.


**Fig. 7.** Prediction results after training using epoch 10.

The MobileNet V2 model employed epoch 10 since it has a smaller loss ratio than other epochs during the training process. Training and validation are compared to select the epoch for producing good accuracy and low loss. Therefore, utilizing epochs beginning with epoch 10 because if fewer than 10 programmers are generated, an error prevents displaying the training results and the ensuing graph since the epoch is too tiny, resulting in underfitting of the data.

Thus, if the number of epochs increases, the weight will change, and the graph curve will be more curved from a less suitable curve to align with an overfitting curve. The number of epochs depends on the dataset. There is no definite reference in determining the epoch value that wants to be applied. This is related to the convergence of the trained neural network. The MobileNet V2 model is considered convergent if and only if the parameter loss during training indicates that there is no longer any significant performance increase during the training process. The value lost during training does not decrease significantly as the epoch increases. It is a sign that the model has attained convergence, at which point it is recommended to either increase the number of epochs or terminate the training process.

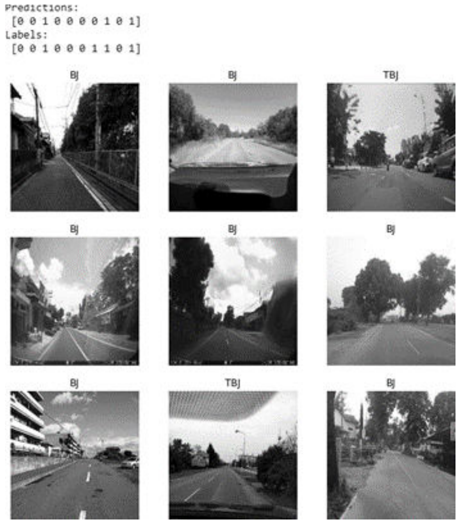


**Fig. 8.** Prediction results after training using 10 batch size.

The results of the prediction image after the training process are carried out using batch size 10, and 10 epochs with an error rate between the label and prediction of 10% as illustrated in Fig 8. where the prediction is the road shoulder image of the training results, and the label is the ground truth.

Based on the results of system validation obtained from the training results, the accuracy and percentage of batch size change errors presented in Table III derived from the training results are significantly more optimal.

**TABLE III.** SYSTEM VALIDATION RESULTS

| **Batch size** | **Accuracy** | **Error** |
|---|---|---|
| 10 | 0.900 | 0.100 |
| 16 | 0.875 | 0.125 |
| 24 | 0.700 | 0.300 |
| 32 | 0.903 | 0.097 |

From Table III, a comparison graph of the confusion matrix variables was obtained as illustrated in Fig 9. a graph comparing the categorization of the road shoulder. Based on the graph and the accuracy obtained from Table 3, less than 10% of the models are less able to detect and mis detect. The MobileNet V2 model obtained was relatively good since when the system validation was carried out, the model's accuracy using manual calculations has an incorrect accuracy that was smaller than the correct accuracy. This can be evidenced by using prediction, as illustrated in Fig. 7 and 8.

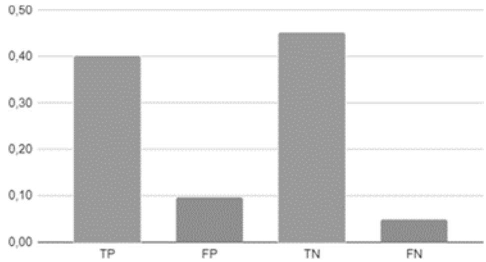


**Fig. 9.** Variable Comparison Graph

## 4. CONCLUSION

The results MobileNet V2 classifying obtained cannot be separated from overfitting and to minimize the resulting error, the values in the hyperparameters are changed, namely Epochs and Batch Size. In the training process, the most optimum parameters were using epochs 10, 20, 30, 40 and Batch sizes 10, 16, 24 with a learning rate of 0.0001 and Optimizer RMSprop in training from the MobilenetV2 model which produces high accuracy of approximately 0.7 to 0.8 and a loss value as minimal as possible with a minimum step of approximately 0.2 to 0.3. The data generated by the training process road shoulder were overfitting at epochs 50 to 100, where the value of training and validation differs in another way, indicating that during the training process, loss and training accuracy results are more significant than loss and validation training results. However, based on additional testing, the forecasts have an error rate of less than 20% and can still be used.

## REFERENCES

1. G. Aptarila and F. Lubis.: "Analisis Kerusakan Jalan Metode SDI Taluk Kuantan - Batas Provinsi Sumatera Barat," Jurnal Teknik Sipil, vol. 6, no. 2, pp. 195-203, (2020).
2. R. Hartono, Y. Wibisono and R. A. Sukamto.: "Damropa (Damage Roads Patrol): Aplikasi Pendeteksi Jalan Rusak Memanfaatkan Accelerometer pada Smartphone," June (2020).
3. N. Wakhidah, S. Asmiatun and A. N. Putri.: "Implementasi Metode Clustering Untuk Pengelompokkan Koordinat Kondisi Jalan Berbasis Google Map Implementation Clustering Method For Grouping Road Condition Coordinates Based On Google Map," pp. 150 - 156, November (2019).
4. K. Doshi and Y. Yilmaz.: "Road Damage Detection using Deep Ensemble Learning," in Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020, December (2020).
5. Wijayanto.: "Banyak Jalan Berlubang, Kemantapan Jalan di Jatim Belum Optimal," 02 March (2020).
6. I. t. K.: "Cepat Rusak, Perkerasan Beton Bahu Jalan Cageur -Ciamis Banyak Retak," Mediakomonline.com,[Online]. Available: http://medikomonline.com/berita/infrastruktur/cepat- rusak--perkerasan-beton-bahu-jalan-cageur-ciamis- banyak-retak. [Accessed (2020)].
7. R. Mudiyono and N. S. Tsani.: "Analisis Bahu Jalan Menggunakan Perkerasan Paving Block," jurnal.untidar.ac.id, vol. 03, no. 02, pp. 42 - 50, (2019).

8. S. N. Purnomo , P. B. Santoso , W. Widiyanto, L. Aulian and S. Gunawan.: "Pendekatan Eksperimen Pada Pola Gerusan Di Bahu Jalan Yang Tidak," Jurnal Teknik Sipil dan Perancanaan , pp. 25 - 30, (2017).
9. H. Maeda, Y. Sekimoto , T. Seto, T. Kashiyama and H. Omata.: "Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone," pp. 4 - 6, (2018).
10. M. T. Cao, N. M. Nguyen and K. T. Chang.: "Survey on performance of deep learning models for detecting road damages," (2020).
11. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and C. L. Chen.: "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2018).
12. I. D. Pratama, H. Mahmudah and R. W. Sudibyo.: "Design and Implementation of Real-time Pothole Detection using Convolutional Neural Network for IoT Smart Environment," Proc. 2021 Int. Electron. Symp. IES pp. 675–679, (2021).
13. Z. S. Hernanda, H. Mahmudah and R. W. Sudibyo.: "CNN-Based Hyperparameter Optimization Approach for Road Pothole and Crack Detection Systems," 2022 IEEE World AI IoT Congress (AIIoT), 2022, pp. 538-543, doi: 10.1109/AIIoT54504.2022.9817316(2022)
14. D. O. R. H. P. I. M. a. B. S. D. R. Sulistyaningrum.: "Pavement Distress Classification Using Deep Learning Method Based on Digital Image," January (2020).
15. M. D. S. K. S. Z. a. C. L. Y. J. Wang.: "Deep Proposal and Detection Network for Road Damage Detection and Classification," in IEE International Conferecnce on Big Data, Big Data (2018, 2019).
16. K. Dong, Y. Ruan, C. Zhou and Y. Li.: "MobileNetV2 Model for Image Classification," in International Conference on Information Technology and Computer Application (ITCA), (2020).
17. M. Akay.: "Deep Learning Classification of Systemic," Engineering in Medicine and Biology," vol. 2, (2021).
18. N. K. B. Sai and S. T.: "Object Detection and Count of Objects in Image," in Second International Conference on Smart Systems and Inventive Technology , (2019).