2022 International Conference on Electrical Engineering, Computer and Information Technology
University of Jember, Jember, Indonesia, November 22-23 2022

Detection Of Road Damage Using Faster Regional-Convolutional Neural Network Method

Rudy Rachman
Dept. of Informatics
Engineering Faculty
Mulawarman University
Samarinda, Indonesia
rudyrachman16@gmail.com

Anindita Septiarini
Dept. of Informatics
Engineering Faculty
Mulawarman University
Samarinda, Indonesia
anindita@unmul.ac.id

Hamdani Hamdani
Dept. of Informatics
Engineering Faculty
Mulawarman University
Samarinda, Indonesia
hamdani@unmul.ac.id

2022 International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT) | 978-1-6654-9352-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEECIT55908.2022.10030629

Abstract— **Road damage is a common occurrence daily. Even in developed nations, road damage is still a possibility. Rainwater, temperature and weather variations, air temperature, construction materials on the road, subgrade conditions on the road, poor compaction process above the subgrade, and vehicle weights that exceed the limit are all factors that cause damage. Potholes are the most common form of road damage. Damaged roads are a major annoyance for drivers, can lead to accidents, and can even result in on-the-spot deaths due to falls since drivers are unaware of the potholes. If road damage is not discovered or overcome, it can be dangerous, and the road will deteriorate. Many media, including images, can be used to detect road damage. This study uses the image by utilizing the Faster R-CNN method to detect road damage. It reveals that using the MobilenetV2 backbone achieved the optimal performance indicated by the mAP value of 79.7%.**

Keywords—Faster Regional Convolutional Neural Network (R-CNN), MobilenetV2, neural network, pothole, road damage

I. Introduction

Road damage can occur anywhere; even in developed countries, damage can still occur. Damage can occur due to various variables, including poor compaction process, quality of asphalt on the road, bad drainage, vehicles weight exceeding the limit, etc., [1]. The most common type of road damage is potholes. The pothole is very disturbing for road users. It can also cause an accident and even die on the spot due to falling because they are unaware of a pothole.

The previous studies look for various ways to detect road damage to make it easier for government agencies or other parties to detect damage to a road even though it was not in place [2][3][4][5][6]. Many media are already used for study, e.g., images and sensors installed in vehicles. Also, using image processing or artificial intelligence methods. Developments in image processing and artificial intelligence are often used to detect objects, e.g., an object on the road, road signs, etc.

Several studies used image processing with 3D images [7] or 2D images [2], [5], [8]. The advancement of image processing technology was used to identify net damage in the water [9]. Image processing was implemented to detect potholes [5]. The development of artificial intelligence methods, namely deep learning carried out to detect the small road sign in the image [10], [11] using Faster Regional-Convolutional Neural Network (R-CNN).

The previous studies that used image processing still have shortcomings, such as poor image conditions due to taking pictures that are reflected in the sun's light, or having shadows, since it is likely that the resulting findings will not be satisfactory [5]. Therefore, several studies have often taken advantage of the development of deep learning, especially Convolutional Neural Network (CNN) [12], because CNN performs better than the image processing method when classifying an object. For detecting an object in an image, researchers use Faster R-CNN for detecting various objects [10], [11], [13][14][15][16]. Faster R-CNN had better performance even if the images were taken inappropriately because it uses the image as training data.

Therefore, this study used Faster R-CNN for detecting road damage using the image as the input. This study only detects potholes in the image. Dataset was provided from a public resource. Furthermore, this study presents the optimal result by modifying the Non-Maximum Suppression (NMS) threshold [17] and switching the backbone of the Region Proposal Network (RPN) in Faster R-CNN.

II. Related Works

In this session, will discuss about previous work of authors that use CNN or Faster R-CNN. Their works become the theoretical basis of this research.

The CNN model was used to classify cat and dog images. This study uses Visual Geometry Group 16 (VGG-16) for the architecture of the CNN model. Datasets consist of 5000 dog and cat images. This study also does data augmentation to make the model more robust when classifying an image [12]. The deep convolutional network was used to identify picture cracks. Datasets were collected primarily using a smartphone camera. There are 500 images with 32642448pixel resolution. This study uses three methods, e.g., SVM, Boosting, and ConvNets. ConvNets, which provide the best precision, recall, and F1 score results, were employed to achieve the best outcome in this investigation [18].

A new classification method called Faster R-CNN was proposed by applying the Pascal VOC and MS COCO datasets for object detection. The Faster R-CNN model detection result was the best of any model tested; each image's detection speed is 200 milliseconds [13]. Faster R-CNN was used to detect a small road sign in an image. The RPN model of the Faster R-CNN model was applied VGG-16 to the improved NMS called soft NMS for detecting the small object. Intersection over Union (IoU) for the positive sample of this study is 0.67 and for the negative example is

0.3. The result showed better detection of minor road signs in an image [11].

Furthermore, Faster R-CNN was implemented to detect faces in the image. The RPN model of the Faster R-CNN model is VGG-16. There were 12,880 images of 159,424 faces used for training the model. The study applied a scheduled learning rate; when the first 50,000 iterations, the learning rate was 0.001, and after that, the subsequent 30,000 iterations, the learning rate was 0,0001. Faster R-CNN results are the best among other models, which also test on the same dataset and the same training time [16].

III. Dataset and Methods

A. *Dataset*

The dataset used in this research is available online. After discarding the lousy images (pothole is too close when taking the image, the image is high contrast, too blurred, too dark, or the pixel too small), the dataset contains 665 images of the pothole. The entire image's label was changed to pothole because the model must detect only potholes. All images were also resized to 300×300pixel because the model's input is that size. The image in the dataset can be seen in Fig. 1.

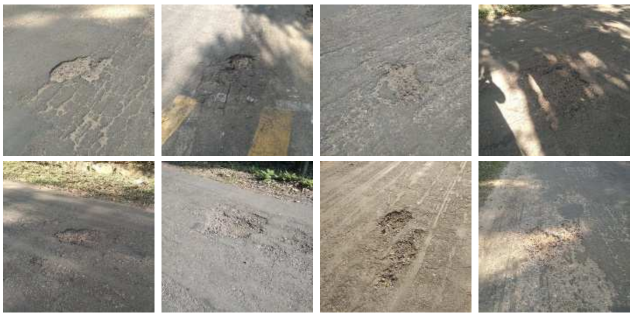


Fig. 1. The road images in dataset

Datasets contain split.json files to split train and test data. This file is created by randomizing the dataset images after the randomization. It needs a few configurations, too, so the split was more complicated when training the model and testing the model will become more robust. Train and test data were 80:20; therefore, the train and test data were 532 images and 133 images, respectively.

After splitting to train and test data, augmentation of the train data was needed to increase the data variation. The main purpose of this augmentation was to increase the number of data, especially on the image. Therefore, augmentation can reduce the occurrence of overfitting [19]. Generally, the augmentation technique used for image classifying was rotated, flipped, and cropped [20]. In this study, the augmentation methods used were rotated and flipped against the train data. There were four augmentation techniques applied, including flip vertical (1)-(2), flip horizontal (3)-(4), rotate 90 to the right (5)-(8), and rotate 90 to the left (9)-(12).

$$y_{1T} = \text{image width} - y_2 \tag{1}$$

$$y_{2T} = \text{image width} - y_1 \tag{2}$$

$$x_{1T} = \text{image height} - x_2 \tag{3}$$

$$x_{2T} = \text{image height} - x_1 \tag{4}$$

$$x_{1T} = \text{image width} - y_2 \tag{5}$$

$$x_{2T} = \text{image width} - y_1 \tag{6}$$

$$y_{1T} = x_1 \tag{7}$$

$$y_{2T} = x_2 \tag{8}$$

$$x_{1T} = y_1 \tag{9}$$

$$x_{2T} = y_2 \tag{10}$$

$$y_{1T} = \text{image height} - x_2 \tag{11}$$

$$y_{2T} = \text{image height} - x_1 \tag{12}$$

Equations (1)-(12) were used for the bounding boxes of the objects. After augmentation, the datasets are exported to the TF-Record file. The file was used to train and test the model with the private dataset. TF-Record files were created separately; two files were used for training and testing.

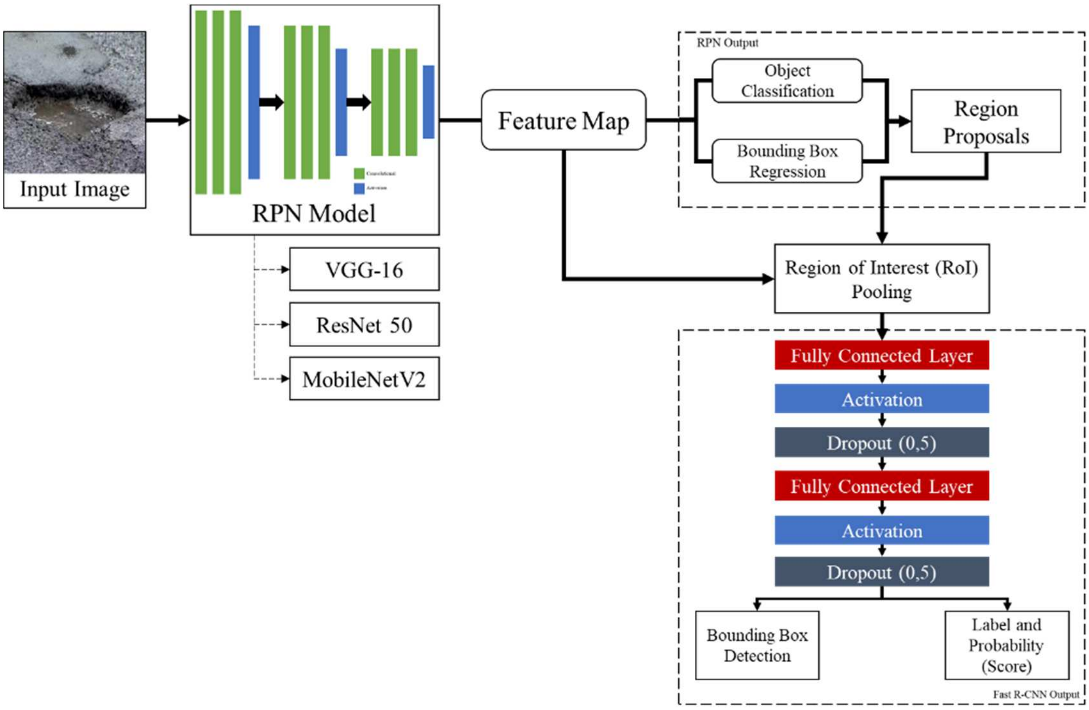


Fig 2. Faster R-CNN model

B. Faster R-CNN

The Faster R-CNN model applied in this study was adopted as in [13]. The difference was the input size and adding another fully connected layer in the end. The input size of this model was 300, 300, and 3. The architecture of the Faster R-CNN model in this study can be seen in **Error! Reference source not found.**. Faster R-CNN was divided into two stages. The first stage was RPN detecting an object by using an anchor. Then, the data from RPN were classified by the detector (Fast R-CNN). The outputs of this model are the bounding box coordinates (y_1, x_1, y_2, and x_2) and the label and probability score of the prediction

RPN models used in this study were VGG-16, MobileNetV2, and Deep Residual Net 50 (ResNet50). RPN model did not take all architecture; this study only took until the output shape was 19×19 for MobileNetV2 and ResNet-50; for VGG-16, the output shape was 18×18. The last output shape of the RPN model will be used as a feature map shape. The feature map will be used as input for Region of Interest (RoI) Pooling and region proposals. In region proposals, the NMS method suppresses object classification and bounding box regression. If the prediction IoU was below the threshold, the prediction was saved. However, if the prediction IoU was above the threshold, the prediction was discarded. NMS method was also applied to the Faster R-CNN predictor (Fast R-CNN) only to show the best prediction bounding box. NMS threshold is the parameter model this study used to find the best result.

This study's optimizer for the Faster R-CNN model is Stochastic Gradient Descent (SGD). The learning rate is set at 0.00092, with momentum set at 0.8. The model also will stop the training if all loss value is lower than 10%. This paper use precision (13), recall (14), and mean Average Precision (mAP) results for testing model (15)-(17) [21]. The overview of the experiment flowchart is seen in Fig. 2.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \times 100 \qquad (13)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \qquad (14)$$

$$\text{AP} = \frac{1}{11} \sum_{R \in \{0,0.1,\ldots,0.9,1\}} P_{\text{interp}}(R) \qquad (15)$$

$$P_{\text{interp}}(R) = \max_{\tilde{R}:\tilde{R} \geq R} P(\tilde{R}) \qquad (16)$$

$$\text{mAP} = \frac{\sum_{k=1}^{N} \text{AP}(k)}{N} \times 100 \qquad (17)$$

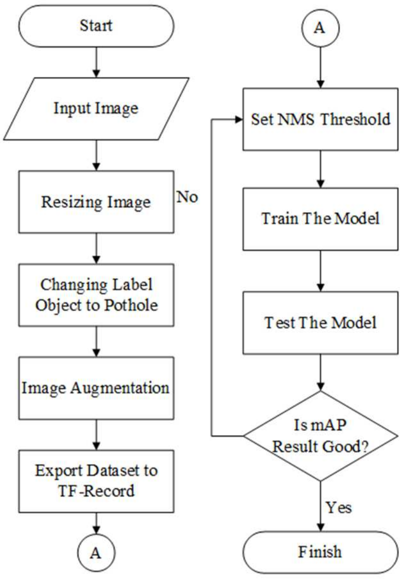


Fig. 2. Experiment flowchart

IV. Result And Discussion

This study implemented the evaluation performance using three parameters: precision, recall, and mAP. After a few pieces of training and testing, this paper will use the NMS threshold. While training is 0.7, as in Ren et al. study [13], the NMS threshold for testing is 0.5. IoU threshold for this paper use 0.5; if the bounding box has an IoU score below it, that bounding box is False Positive (FP), but if above the threshold score, that bounding box is True Positive (TP). Fig. 3(a) is called image A, Fig. 3(b) is called image B, and Fig. 3(c) is called image C.

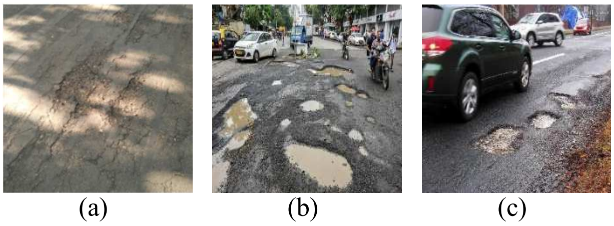


Fig. 3. Testing image for precision and recall

The prediction results were achieved utilizing three different backbones (VGG-16, MobileNetV2, and ResNet-50) and images A, B, and C, as shown in Figure 5. There are two bounding boxes: green boxes represent the ground truth, and red boxes represent model predictions. The prediction result of image A in Fig. 4 proves that the backbone using VGG-16 could not detect a pothole covered by shadow well, but MobilenetV2 and ResNet-50 could predict each hole correctly. The precision and recall results of image prediction A can be seen in TABLE I. for VGG-16, TABLE II. for MobilenetV2, and TABLE III. for ResNet-50.

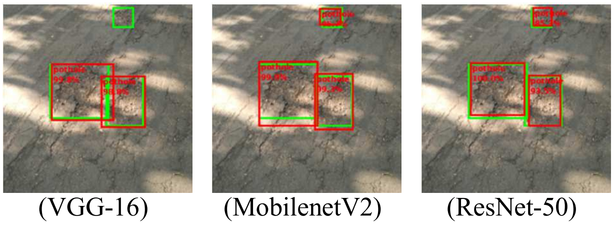


Fig. 4. Prediction result image A

TABLE I. PRECISION AND RECALL RESULT OF IMAGE A WITH VGG-16

Detection	Precision	Recall
TP	100%	33.3%
TP	100%	66.7%

TABLE II. PRECISION AND RECALL RESULT OF IMAGE A WITH MOBILENETV2

Detection	Precision	Recall
TP	100%	33,3%
TP	100%	66,7%
TP	100%	100%

TABLE III. PRECISION AND RECALL RESULT OF IMAGE A WITH RESNET-50

Detection	Precision	Recall
TP	100%	33,3%
TP	100%	66,7%
TP	100%	100%

The prediction results of image B in Fig. 5 prove that the three models are not able to detect potholes if the water holes are completely covered with water. The potholes that are too small and too far from the image acquisition distance also make the model not recognize whether there is damage or not. The results of B image prediction precision and recall can be seen in TABLE IV. for VGG-16, TABLE V. for MobilenetV2, and TABLE VI. for ResNet-50.

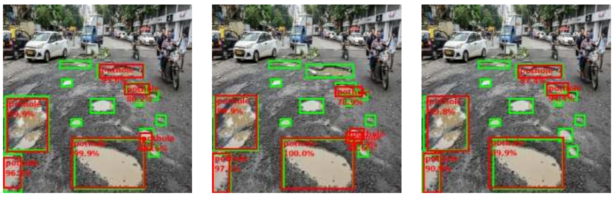

(VGG-16) (MobilenetV2) (ResNet-50)

Fig. 5. Prediction Result Image B

TABLE IV. PRECISION AND RECALL RESULT OF IMAGE B WITH VGG-16

Detection	Precision	Recall
TP	100%	7.1%
TP	100%	14.2%
TP	100%	21.4%
TP	100%	28.5%
TP	100%	35.7%
FP	83.3%	35.7%

TABLE V. PRECISION AND RECALL RESULT OF IMAGE B WITH MOBILENETV2

Detection	Precision	Recall
TP	100%	7.1%
TP	100%	14.2%
TP	100%	21.4%
TP	100%	28.5%
FP	80%	28.5%
FP	66.7%	28.5%

TABLE VI. PRECISION AND RECALL RESULT OF IMAGE B WITH RESNET-50

Detection	Precision	Recall
TP	100%	7,1%
TP	100%	14,2%
TP	100%	21,4%
TP	100%	28,5%
TP	100%	35,7%

The prediction result of the C image in Fig. 6 proves that the backbone with the ResNet-50 model has a reasonably high sensitivity. The ResNet-50 model detects fairly dark road edges that look like potholes, and predictions that are like small, waterless potholes are also predicted, but they are not listed as potholes in the ground truth. The precision and recall results of C image prediction can be seen in TABLE VII. for VGG-16, TABLE VIII. for MobilenetV2, and TABLE IX. for ResNet-50.

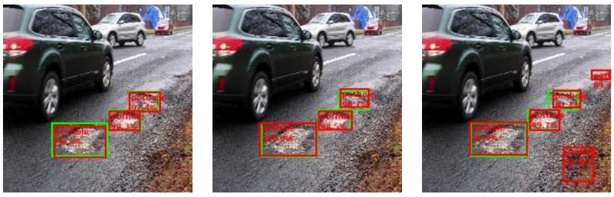

(VGG-16) (MobilenetV2) (ResNet-50)

Fig. 6. Prediction result image C

TABLE VII. PRECISION AND RECALL RESULT OF IMAGE C WITH VGG-16

Detection	Precision	*Recall*
TP	100%	33,3%
TP	100%	66,7%
TP	100%	100%

TABLE VIII. PRECISION AND RECALL RESULT OF IMAGE C WITH MOBILENETV2

Detection	Precision	Recall
TP	100%	33,3%
TP	100%	66,7%
TP	100%	100%

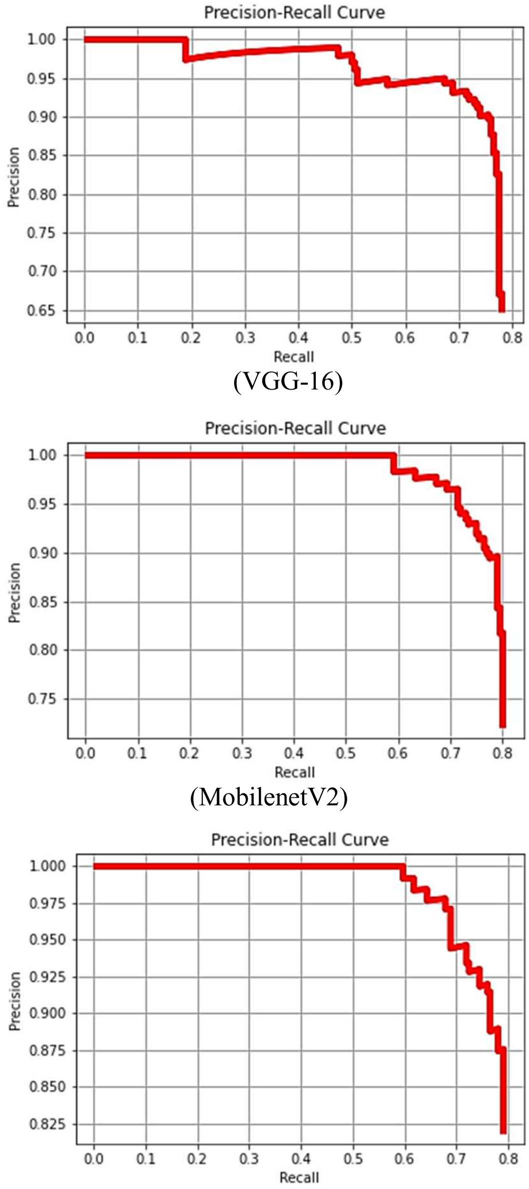
TABLE IX. PRECISION AND RECALL RESULT OF IMAGE C WITH RESNET-50

Detection	Precision	Recall
TP	100%	33,3%
TP	100%	66,7%
TP	100%	100%
FP	75%	100%
FP	60%	100%

The testing result for mAP, precision, and recall curve images were carried out using 133 test data images. The results were obtained using three different backbones, such as testing with three images. The mAP results for each backbone can be seen in TABLE X. and the precision and recall curves can be seen in Fig. 7.

TABLE X. MAP RESULT

Backbone	Total Epoch	mAP
VGG-16	31	71,1%
ResNet-50	29	72,1%
MobilenetV2	67	79,7%



(VGG-16)

(MobilenetV2)

(ResNet-50)

Fig. 7. Precision and Recall Curve

The three images in Fig. 4, image B, is the image that has the False Negative conditions frequently. This condition occurs because the model can still not detect potholes where the holes are filled with water, holes filled with water, and too small or reflected by light; the shooting distance is too far, so the holes are not like the potholes by the model. The most satisfactory detection result from image B is ResNet-50 because it has the best precision and recall values among other models.

In contrast to image B, image A has a lower difficulty detecting. The model that uses the VGG-16 architecture fails to detect all the damage. It caused the VGG-16 model could not notice if the hole was small and covered in shadows. The most satisfactory detection results on image A of the three models are MobilenetV2 and ResNet-50 because these models can detect all defects well; the precision and recall values are also the same, but for the detection score, MobilenetV2 is more convincing than ResNet-50.

Finally, image C is the image that has the most accessible level of difficulty. All models could detect well except ResNet-50 because ResNet-50 creates a False Positive for the roadside with a few black spots and looks like a small pothole at the end of the image. The most satisfactory detection results on image C are the VGG-16 and MobilenetV2 models because both models could detect all types of damage and do not make False Positives like ResNet-50. The precision and recall results are also the same, but VGG-16 has a more convincing score for damage detection confidence than MobilenetV2.

The results of testing the mAP values of the three RPN architectural models show that the model using MobilenetV2 as the backbone has the highest mAP value of 79.7%, compared to other models, ResNet-50 only 72.1%, and VGG-16 the lowest at 71.1%. MobilenetV2 has the highest mAP value, but the number of epochs used to train the model with the MobilenetV2 backbone is very large. The model required 67 epochs, compared to VGG-16s of 31 epochs, and ResNet-50s had the least of 29 epochs.

V. Conclusion

The conclusion that can be drawn from this study is that the Faster R-CNN model utilizing the MobilenetV2 backbone achieved the highest success rate, 79.7% when compared to other models. Still, the training process is 67 epochs, compared to using VGG-16 which has the mAP value of 71.1% with a total epoch of 31, or ResNet-50 with the mAP value of 72.1% with a total epoch of 29. The model still cannot detect potholes filled with water; the pothole is too small or too far from the image acquisition distance due to a lack of training data for the case.

VI. Future Work

On our next research, we will make sure can improve model result by maintaining the dataset of image to train the model by collecting and create own dataset of pothole or another road damage image. Add more augmentation method to train the model. Also, we have plan to implement it become real time detector.

References

[1] S. Ibnu and U. Nugroho, "Road Damage Analysis of Kalianak Road Surabaya," *Adv Sci Lett*, vol. 23, no. 12, pp. 12295–12299, Dec. 2017, doi: 10.1166/asl.2017.10624.

[2] L. Huidrom, L. K. Das, and S. K. Sud, "Method for Automated Assessment of Potholes, Cracks and Patches from Road Surface Video Clips," *Procedia Soc Behav Sci*, vol. 104, pp. 312–321, 2013, doi: 10.1016/j.sbspro.2013.11.124.

[3] H. Majidifard, Y. Adu-Gyamfi, and W. G. Buttlar, "Deep machine learning approach to develop a new asphalt pavement condition index," *Constr Build Mater*, vol. 247, p. 118513, 2020, doi: 10.1016/j.conbuildmat.2020.118513.

[4] F. Meng and A. Li, "Pavement Crack Detection Using Sketch Token," *Procedia Comput Sci*, vol. 139, pp. 151–157, 2018, doi: 10.1016/j.procs.2018.10.231.

[5] T. Siriborvornratanakul, "An Automatic Road Distress Visual Inspection System Using an Onboard In-Car Camera," *Advances in Multimedia*, vol. 2018, 2018, doi: 10.1155/2018/2561953.

[6] C. van Geem *et al.*, "Sensors on Vehicles (SENSOVO) - Proof-of-concept for Road Surface Distress Detection with Wheel Accelerations and ToF Camera Data Collected by a Fleet of Ordinary Vehicles," *Transportation Research Procedia*, vol. 14, pp. 2966–2975, 2016, doi: 10.1016/j.trpro.2016.05.419.

[7] L. Inzerillo, G. di Mino, and R. Roberts, "Image-based 3D reconstruction using traditional and UAV datasets for analysis of road pavement distress," *Autom Constr*, vol. 96, no. May, pp. 457–469, 2018, doi: 10.1016/j.autcon.2018.10.010.

[8] S. K. Ryu, T. Kim, and Y. R. Kim, "Image-Based Pothole Detection System for ITS Service and Road Management System," *Math Probl Eng*, vol. 2015, 2015, doi: 10.1155/2015/968361.

[9] Y. P. Zhao, L. J. Niu, H. Du, and C. W. Bi, "An adaptive method of damage detection for fishing nets based on image processing

technology," *Aquac Eng*, vol. 90, no. March, p. 102071, 2020, doi: 10.1016/j.aquaeng.2020.102071.

[10] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1951–1959, 2017, doi: 10.1109/CVPR.2017.211.

[11] C. Cao *et al.*, "An Improved Faster R-CNN for Small Object Detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019.2932731.

[12] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[14] K. bing Chen, Y. Xuan, A. jun Lin, and S. hua Guo, "Esophageal cancer detection based on classification of gastrointestinal CT images using improved Faster RCNN," *Comput Methods Programs Biomed*, vol. 207, p. 106172, Aug. 2021, doi: 10.1016/j.cmpb.2021.106172.

[15] Q. Zhang, X. Chang, Z. Meng, and Y. Li, "Equipment detection and recognition in electric power room based on faster R-CNN," *Procedia Comput Sci*, vol. 183, pp. 324–330, Jan. 2021, doi: 10.1016/J.PROCS.2021.02.066.

[16] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, pp. 650–657, 2017, doi: 10.1109/FG.2017.82.

[17] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6469–6477, 2017, doi: 10.1109/CVPR.2017.685.

[18] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2016-Augus, pp. 3708–3712, 2016, doi: 10.1109/ICIP.2016.7533052.

[19] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," Dec. 2017, Accessed: Aug. 03, 2021. [Online]. Available: https://arxiv.org/abs/1712.04621v1

[20] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *2017 Chinese Automation Congress (CAC)*, Oct. 2017, pp. 4165–4170. doi: 10.1109/CAC.2017.8243510.

[21] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July, no. July, pp. 237–242, 2020, doi: 10.1109/IWSSIP48289.2020.9145130.