International Journal of GEOMATE, July., 2021, Vol.21, Issue 83, pp.-87-94
ISSN: 2186-2982 (P), 2186-2990 (O), Japan, DOI: https://doi.org/10.21660/2021.83.6153
Geotechnique, Construction Materials and Environment

DEFECT DETECTION ON ASPHALT PAVEMENT BY DEEP LEARNING

Jonpaul Nnamdi Opara[1], Aunt Bo Bo Thein[2], Shota Izumi[3], Hideaki Yasuhara[4], and *Pang-Jo Chun[5]

[1,2,3,4] Graduate School of Science and Engineering, Ehime University, Japan; [5] School of Engineering, The University of Tokyo, Japan.

*Corresponding Author, Received: 28 Nov. 2020, Revised: 22 Mar. 2021, Accepted: 04 Apr. 2021

ABSTRACT: The importance of road infrastructure to the economy of any nation cannot be overemphasized, however, it is not easy to maintain it properly, the increase in maintenance and repair expenditures are issues of concern coupled with the constantly increasing number of roads. Since the inspection of pavements is particularly difficult, an efficient inspection method is required. In this study, a method for detecting damage in asphalt pavements was developed using one of the deep learning techniques, YOLOv3. YOLOv3 is a method for detecting the position and type of an object from an input image, which fits the purpose of this study. The developed method can distinguish between longitudinal crack, transverse crack, alligator crack, and pothole. To confirm the accuracy of the developed method, images of pavements acquired on National Route 4 using were analyzed. From the analysis, it is found that the precision value is 0.7 and the average IoU is 50.39%. From the visualization of the analysis results, it was found that this method based on YOLOv3 was able to detect the damage with good accuracy. This is a significant improvement and can help shape the entire road Inspection procedures.

Keywords: Asphalt pavement, Deep learning, YOLOv3, Crack detection, Pothole

1. INTRODUCTION

The importance of road infrastructure to the economy of any nation cannot be overemphasized [1]. Roads are the arteries through which the economy pulses. By linking producers to markets, workers to jobs, students to school, and the sick to hospitals, roads are vital to any development agenda. Countries all over the world are currently finding the path towards attaining the Sustainable Development Goal 9 (SDG9), efficient transportation services are key drivers of economic development [2]. Road's importance s infinite, but most importantly is how safe the road is and the ease of maintenance that matters. No matter the amount of road constructed, proper maintenance is essential to ensure that the road continues to serve its design purpose. In order to achieve this, a robust road management system has become a necessity. Road inspection is one of the key processes of a pavement management system, whose function is to examine and describe the road infrastructure's condition [3]. If done right will provide adequate information required to implement a robust and all-encompassing road policy and maintenance schedule.

In Japan for instance, amid the aging of infrastructures, the increase in maintenance and repair expenditures are issues of concern. In a 2016 survey, the total length of roads in Japan is about 1,210,000 kilometers and the number of roads that require maintenance is constantly on the increase [4]. Among these, 77.3% of the total road length is in asphalt concrete structures. This means the inspection and maintenance processes must be cost-effective and efficient to carter for this volume work. In this study, we focus on road inspection, utilizing deep learning techniques.

2. RESEARCH SIGNIFICANCE

Achieving the goal of this study will be a very significant endeavor in improving defects detection on asphalt pavement the entire road maintenance and road safety assurance process at large when compared to the current methods available. It is expected that the findings will help in improving and providing a real-time and automated approach for defect detection, improved accuracy "free from unnecessary human error" and reduced road inspection time. The study findings and data if positive, will also, afford future researchers' opportunity to gain a deeper understanding of asphalt pavement performance, serving as a platform that can be integrated into the building road inspection models with sensitivity and specificity as the goal.

Looking at the traditional evaluation method which is usually done by human expert, is a process that always involves a certain degree of subjective judgment by the expert. To minimize the subjectivity, save time and labour cost, a computerized system for analysis that automatically evaluates the condition of the pavement is desired.

Many studies have been conducted to automatically detect cracks in asphalt pavements (e.g. [5-11]). If we include the detection of damages in concrete or steel structure, there are many papers on this “Defect Detection” subject including authors’ research (e.g. [12-20]). However, considering that almost all the methods have not yet been put to practical use, there’s the need to accumulate more research developments. Therefore, in this study, we developed a method for detecting asphalt pavement using YOLOv3 [21], which is a deep learning technique to detect the position and type of an object from an input image. Then, images of pavements acquired on National Route 4 in Japan were analyzed to confirm the accuracy of the developed method. We will discuss these in the following chapters.

3. LITERATURE REVIEW ON CRACK DETECTION IN ASPHALT PAVEMENT

With a lot of research in developing defect detection and inspection system from the survey database images. Deep learning and AI applications have become vital for maintenance infrastructure management. And lots of advancements been made in this area especially with the application of computer vision techniques. [22] identified Faster R-CNN as a better algorithm in terms of speed and proposed a solution to detect road potholes, deep ridges, and speed breakers with an R-CNN-based model. The proposed approach used the custom-built Bumpy Dataset which was created, both the trained model and the dataset used to develop the model could easily be integrated into an assistive driving technology.[23] made a comparative study on automatic asphalt pavement cracks recognition based on image processing and machine learning approaches. Six machine learning approaches, Naïve Bayesian Classifier (NBC), Classification Tree (CT), Backpropagation Artificial Neural Network (BPANN), Radial Basis Function Neural Network (RBFNN), Support Vector Machine (SVM), and Least Squares Support Vector Machine (LSSVM), have been employed. Additionally, Median Filter (MF), Steerable Filter (SF), and Projective Integral (PI) have been used to extract useful features from pavement images.

The findings showed that input pattern including the diagonal PIs enhances the classification performance significantly by creating more informative features. LSSVM had the highest classification accuracy, which indicates that machine learning can in understanding pavement conditions properly. Although in this research we dived in deeper, works like this set the bases for more research.

In [24] the authors narrowed down to crack defects on pavement and propose the CrackSegan end-to-end trainable deep C-NN for pavement crack detection, that has proven to be effective in achieving pixel-level, and automated detection via high-level features. Novel multiscale dilated convolutional module that can learn rich deep convolutional features was introduced, making the crack features acquired under a complex background more discriminant. Using their own custom CrackDataset to train and evaluate the CrackSeg net the experimental results showed that the CrackSeg can achieve high performance with a precision of 98.00%, recall of 97.85%, F-score of 97.92%, and a mIoU of 73.53%. Comparing this with other state-of-the-art methods, the CrackSeg performs more efficiently, and robustly for automated pavement crack detection and could be deployed for other forms of pavement defect detection.

4. CRACK DETECTION METHOD

Recently, many deep learning techniques have been developed. Among them, YOLOv3 is positioned as extremely fast and accurate object detection algorithm. YOLOv3 makes use only features learned by a deep convolutional neural network to detect an object, making it a fully convolutional network (FCN), the algorithm classifies objects into a category, it can also detect multiple objects within an image outputting the 4-coordinate and the bounding boxes for each object. In this study, we aimed to detect four types of defects: longitudinal crack, transverse crack, alligator crack and pothole. Examples of each are shown in Fig. 1. To detect and classify them by YOLOv3, the training dataset were made using the annotation tools LabelImg [25]. An example of annotation is shown in Fig. 2.

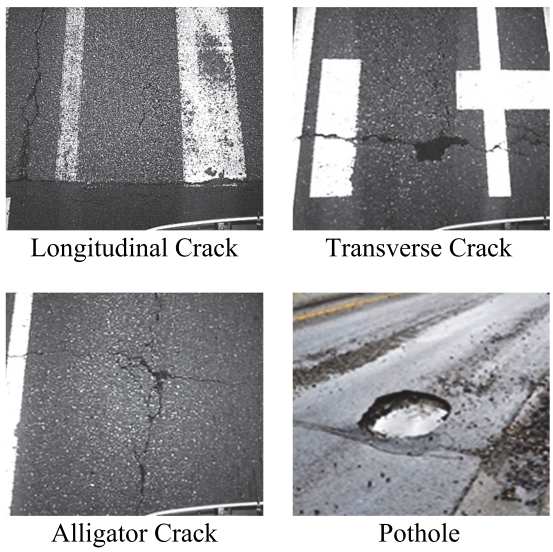

Fig. 1 Classes of asphalt pavement defects

Fig. 2 Example of annotation with LabelImg. (The text in the image is labeled “labelImg”.)

Fig. 3 RIM vehicle owned by Obayashi Road Cooperation.

4.1 Hyperparameters

Hyperparameters are the variable which control the network architecture and training process of YOLOv3. Hyperparameter can be changed according to the own network architecture. In our defects detection model the hyperparameters are set as below. In this research, we use the network size of 448 x 448 with 3 channels. The momentum and decay rate are set to 0.9 and 0.005 respectively. Saturation and Exposure are set to 1.5. Learning rate of the YOLOv3 network is set to 0.001 burning in at 1000 training steps. Policy is set up to do in step by step. As our crack detection network has 4 classes, we use the YOLOv3 layers with 27 filters using linear activation function. Ignore threshold is set to 0.7. We use 9 anchors cluster for the network which is calculated by using computing function presented in latest YOLOv3 repository. We adjust the anchors value and test the training. Maximum iteration is set to 20000 iteration. Usually sufficient 2000 iterations for each class (object), but not less than number of training images and not less than 6000 iterations in total. As our network has 4 classes, we only need about 8000 iteration. However, for the better performance and for the research, we train up to 20000 iteration.

5. DATA COLLECTION

In this research, survey data were taken by a survey vehicle called RIM (Road space Information Management system), owned by Obayashi Road Cooperation (Fig. 3). It is a vehicle mounted with GPS instruments with a mobile mapping system MMS with surface recording cameras which collect the road surface information up to mm units in three-dimensional perspective. The RIM vehicles enable highly detailed mapping of roads and high accuracy and excellent positioning, within a short period of time while minimizing nuisance to traffic and risks of accidents. The image data taken by the RIM vehicle are 2400 x 2000 pixels images and the light content measurement is in RGB. In this study, the survey data is taken along the Nation Highway Road No.4 inside Tochigi Prefecture.

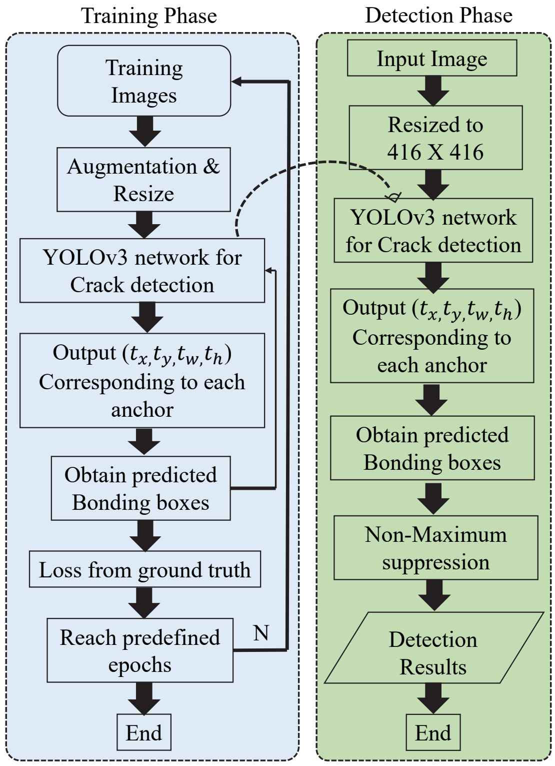

Fig.4 This is the example for figure formatting

Table 1 The number of images for each class

Defect class	The number of pictures
Transverse crack	1,035
Longitudinal crack	1,676
Alligator crack	672
Pothole	11
No crack	1,968

Table 1 shows the number of images obtained for each class. Although the number of potholes is small, it is inevitable because potholes are quickly repaired in Japan. In the future, it is advised that more pothole images are required to improve the detection performance of potholes, but the methodology is still under consideration.

The training process of our YOLOv3 algorithm layers includes as shown in Fig.4 weights file or our detection network is modified in every training

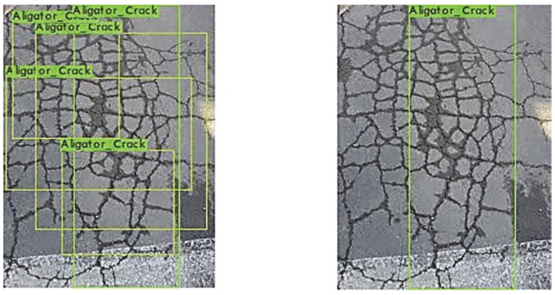

Fig.5 The demonstration diagram of function of Non-maximum suppression

epochs by the loss function which compare the prediction result of the current epochs to ground truth boxes. A method called non-maximum suppression (NMS) is applied to choose the right bounding box among many predictions. The first step of NMS is to suppress all the predictions boxes where the confidence score is under a certain threshold value as shown in Fig 5. The rest of the higher confidence scores are sorted from the highest to the lowest one, then highlight the bounding box with the highest score as the proper bounding box, and after that find all the other bounding boxes that have a high IOU (intersection over union) with this highlighted box. This method allows us to output only one proper bounding box for a detected object. Repeat this process for the remaining bounding boxes and always highlight the highest score as an appropriate bounding box.

YOLO used sum-squared error SSE for the loss function because it is easy to optimize as follows:

$$L = \boldsymbol{\lambda_{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[(\boldsymbol{x_i^j} - \boldsymbol{\hat{x}_i^j})^2 + (\boldsymbol{y_i^j} - \boldsymbol{\hat{y}_i^j})^2\right] + \boldsymbol{\lambda_{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[(\sqrt{\boldsymbol{\omega_i^j}} - \sqrt{\boldsymbol{\hat{\omega}_i^j}})^2 - (\sqrt{\boldsymbol{h_i^j}}\sqrt{\boldsymbol{\hat{h}_i^j}})^2\right] - \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[(\boldsymbol{\hat{c}_i^j}\, \boldsymbol{log}(\boldsymbol{c_i^j}) + (\mathbf{1} - \boldsymbol{\hat{c}_i^j})\boldsymbol{log}(\mathbf{1} - \boldsymbol{c_i^j})\right] - \boldsymbol{\lambda_{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[(\boldsymbol{\hat{c}_i^j}\, \boldsymbol{log}(\boldsymbol{c_i^j}) + (\mathbf{1} - \boldsymbol{\hat{c}_i^j})\boldsymbol{log}(\mathbf{1} - \boldsymbol{c_i^j})\right] - \sum_{i=0}^{s^2} \boldsymbol{l_{ij}^{obj}} \sum_{c\in class} \left[\boldsymbol{\hat{p}_i^j(c)}\, \boldsymbol{log}(\boldsymbol{p_i^j(c)} + \left(\mathbf{1} - \boldsymbol{\hat{p}_i^j(c)}\right) \boldsymbol{log}(\mathbf{1} - \boldsymbol{p_i^j(c)})\right] \quad (1)$$

where λ_{coord} indicates the weight of coordinate error and λ_{noobj} denotes the weight of intersection over union (IoU) error. In this experiment, λ_{coord} equals 5 and λ_{noobj} is 0.5. Next, s^2 is the number of cells ($s \times s$) of feature map. Additionally, $\boldsymbol{l_{ij}^{obj}}$ is used to determine whether the j-th bounding box of the i-th cell is responsible for detecting this object. If the IoU of this bounding box and ground truth is the largest, $\boldsymbol{l_{ij}^{obj}}$ is 1, otherwise, it will be 0.

Similarly, $\boldsymbol{l_{ij}^{noobj}}$ means that the j-th bounding box of the i-th cell is not responsible for the target. Note that cij is the confidence of predicting objects. Additionally, (c) denotes the probability of the class c object in the i-th cell. Because of the use of binary cross-entropy loss and logistic regression for category prediction, this choice helps YOLOv3 to be applied in more complex areas and detect objects more accurately and effectively. Furthermore, it is possible to classify multiple labels on the same object as well. The loss function can be classified as follow:

• The first two terms represent the localization loss in sum-square error equation.
• Terms 3 & 4 represent the confidence loss in sum-square error equation.
• The last term represents the classification loss

The first term:

$$\boldsymbol{\lambda_{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[\left(\boldsymbol{x_i^j} - \boldsymbol{\hat{x}_i^j}\right)^2 + \left(\boldsymbol{y_i^j} - \boldsymbol{\hat{y}_i^j}\right)^2\right] \quad (2)$$

This is a sum square error between the predicted boxes coordinates ($\boldsymbol{x_i^j}$, $\boldsymbol{y_i^j}$) and the ground truth coordinates ($\boldsymbol{\hat{x}_i^j}$, $\boldsymbol{\hat{y}_i^j}$). It is the sum over all the grid cells in the image and for each cell and for each bounding box. Since we have bounding boxes for each cell we need to choose one of them for the loss and this will be the box that has the highest IOU with the ground truth box so the loss will penalize localization loss if that box is responsible for the ground truth box [26]. To comply with that, YOLO uses a binary variable $\boldsymbol{l_{ij}^{noobj}}$, so that $\boldsymbol{l_{ij}^{obj}}$=1 if an object appears in cell i + box j and the cell is responsible for that object, otherwise it 0. $\boldsymbol{l_{ij}^{obj}}$ =1 only if the box contains an object and responsible for detecting this object (higher IoU). The box is responsible for detecting an object if it has the higher IoU with the ground truth box between the B boxes. Since sum square error weights localization error equally with classification error which may not be ideal, YOLO uses a constant (λ_{coord}) to give the localization error a higher weight in the loss function.

The second term:

$$\boldsymbol{\lambda_{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \left[(\sqrt{\boldsymbol{\omega_i^j}} - \sqrt{\boldsymbol{\hat{\omega}_i^j}})^2 - \sqrt{\boldsymbol{h_i^j}}\sqrt{\boldsymbol{\hat{h}_i^j}})^2\right] \quad (3)$$

Here also is similar to the first term, but it calculates the error in the box dimensions. Since the classes of objects thatYOLOv3 can detect has different sizes and Sum-squared error weights errors in large boxes and small boxes equally. Our error metric should reflect those small deviations in large boxes matter less than in small boxes. To partially address this YOLOv3 predicts the square root of the bounding box width and height instead of the width and height directly.

The third term:

$$\sum_{i=0}^{s^2}\sum_{j=0}^{B} l_{ij}^{obj}\left[(\hat{c}_i^j \log(c_i^j) + (1-\hat{c}_i^j)log(1-c_i^j)\right] \quad (4)$$

This is the confidence error where: $0 \leq c_{ij} \leq 1$

The fourth term:

$$\cdot\, \boldsymbol{\lambda_{noobj}} \sum_{i=0}^{s^2}\sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}}\left[(\hat{\boldsymbol{c}}_i^j \log(\boldsymbol{c}_i^j) + (1-\hat{\boldsymbol{c}}_i^j)\boldsymbol{log}(1-\boldsymbol{c}_i^j)\right] \quad (5)$$

If there is no object in the grid we don't need to care about the classification and the localization error. All it needs to care about is the confidence cij (it needs our confidence to be zero when there is no object) and for that YOLOv3 uses variable. $l_{ij}^{noobj}=$ 1 if (there is no object inside cell i) or (there is an object, but the box j for this cell is not responsible for that object), otherwise 0. Since many grid cells do not contain any object, this pushes the confidence scores of those cells towards zero which is the value of the ground truth confidence. This can lead the training to diverge early. To remedy this, we decrease the loss from confidence predictions for boxes that do not contain objects using the parameter λ_{noobj} =0.5.

The last term:

$$\sum_{i=0}^{s^2} l_{ij}^{obj}\sum_{c\in class}\left[\hat{p}_i^j(c)\log(p_i^j(c) + \left(1-\hat{p}_i^j(c)\right)\log(1-p_i^j(c))\right] \quad (6)$$

Here is the binary cross entropy error equation for classification. We summed the errors for all the classes' probabilities for all the grid cells.

In this chapter, we train YOLOv3 using the data collected in the previous chapter. In this study, the ratio of training, validation and test were 7:2:1. In order to prevent overfitting, the training was conducted to an extent that the accuracy of the training dataset and the accuracy of the validation dataset did not deviate [26]. During the training process, precision, recall, and F1 value of validation dataset are obtained. These indices are defined by the following equations.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

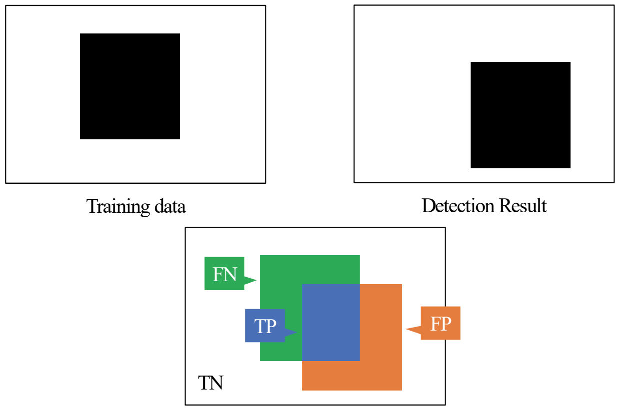

Fig. 6. Definition of *TP, TN, FP, FN.*

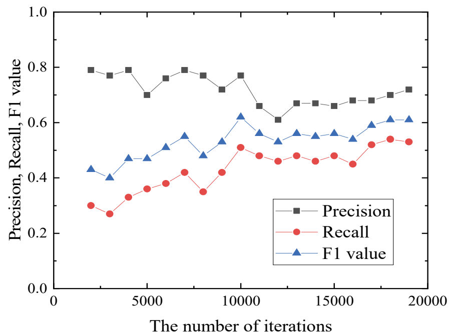

Fig. 7. Changes in precision, recall, F1 value

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{F1 value} = \frac{2\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}} \quad (9)$$

where *TP, TN, FP, FN* are defined as like in Fig.6. Fig. 7. shows the changes in these indices with the number of iteration increases.

5.1 Analyzing Test Images

Figs. 8 to 10 show the detected images obtained in this research. According to the results, the maximum value of precision is at iteration 2000 and 7000, that of recall is at iteration 18000, and that of F1 value is almost at iteration 10000, 18000, and 19000. Considering the above, we decided to adopt the model using the results of iteration 18000 as the analysis model.

Fig. 8 illustrates some examples of the detection results of transverse crack obtained by applying the proposed method. As in the figures, the transverse cracks are accurately detected. Similarly, longitudinal cracks and alligator cracks are also accurately detected as in Fig. 8 and Fig. 9, respectively.

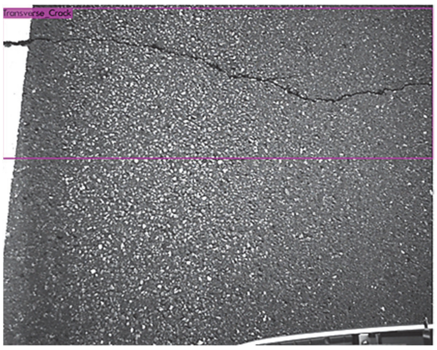

Fig. 8. Detection results of transverse crack
(The text on the purple background in the image is labeled "Transverse Crack")

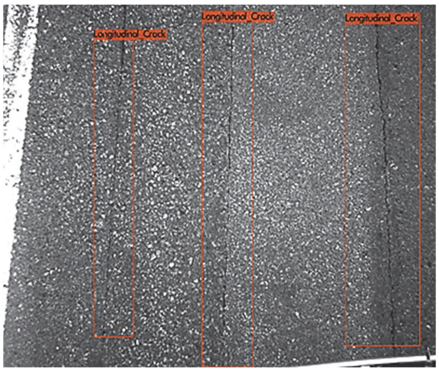

Fig. 9. Detection results of longitudinal crack.
(The text on the orange background in the image is labeled "Longitudinal Crack")

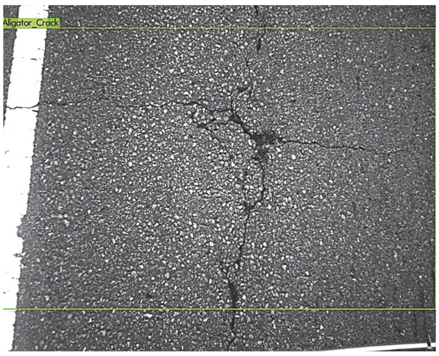

Fig. 10. Detection results of alligator crack.
(The text on the green background in the image is labeled "Alligator Crack")

For the pothole, because there were only a few test images, it is difficult to say that the accuracy is confirmed. However, pothole defect was properly detected on one image as seen in Fig. 11. Though, it is important to re-examine after the number of images including potholes increases in a future study. Fig. 12 shows an example of the detection results when several types of cracks are present in the same image. The top and middle pictures were properly detected, but the bottom picture was

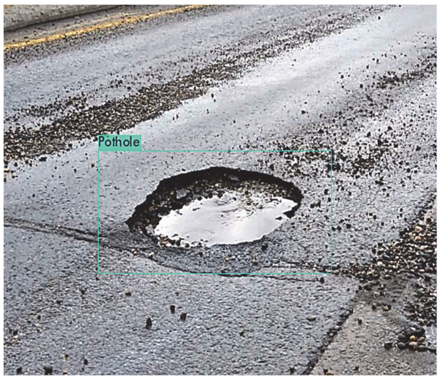

Fig. 11. Detection results of pothole. (The text on the blue background in the image is labeled "Pothole")

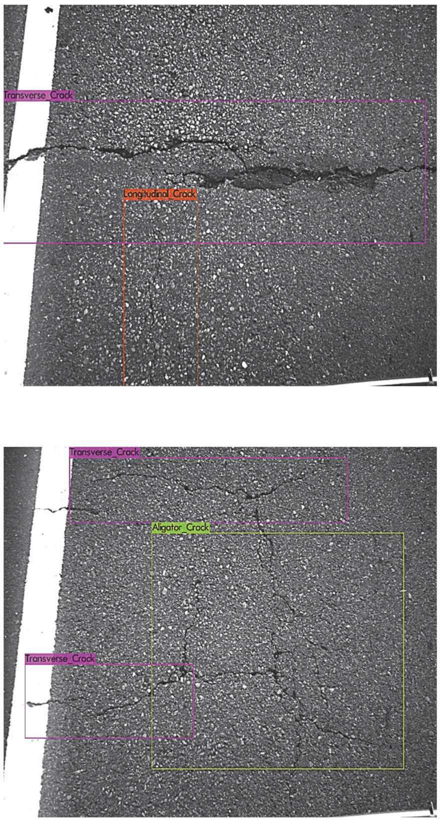

Fig. 12. Detection results when multiple types of cracks are in one image. (The text on the purple background in the image is labeled Transverse Crack, the text on the orange background is labeled Longitudinal Crack, and the text on the green background is labeled Alligator Crack.)

confused between alligator crack and transverse crack. We would like to solve this problem by increasing the amount of data. Nevertheless, this is not a bad result in terms of the detection of cracks.

Considering the results from Fig.8 to Fig.12, it can be said that the effectiveness of this method has been proved.

6. CONCLUSIONS

This study uses YOLOv3, one of the deep learning techniques, to detect cracks in pavements automatically. In addition, the pavement of the National Road was photographed using RIM and analyzed by YOLOv3 to examine the crack detection performance. As a result, it became clear that the crack could be detected with very high accuracy.

Future problems are described below. When there are multiple types of cracks in one image, there is a case in which the exact answer cannot always be obtained. This can be solved by increasing the number of training data. It is also necessary to increase the number of images for the pothole. However, it is generally appropriate as a technique, and it is considered to be one step forward for practical use.

7. ACKNOWLEDGMENTS

We would like to take this opportunity to thank Obayashi Road Corporation for taking the image with their RIM.

8. REFERENCES

[1] Ivanova, E., and Masarova, J., Importance of road infrastructure in the economic development and competitiveness. Economics and management, Vol. 18, Issue 2, pp.263-274, 2013.

[2] Claudia, B., Uwe, D., and Harris, S., How roads support development, World bank blogs, https://blogs.worldbank.org/developmenttalk/ how-roads-support-development (accessed Jun. 29, 2020).

[3] Staniek, M., Detection of cracks in asphalt pavement during road inspection processes, Scientific Journal of Silesian University of Technology. Series Transport., Vol. 96, pp.175-184, 2017.

[4] Ministry of Land, Infrastructure, Transport and Tourism, The maintenance of national road network in Japan, https://www.mlit.go.jp/road/ road e/03key_challenges/1-2-2.pdf (accessed Jun. 29, 2020).

[5] Hoang, N. D. and Nguyen, Q. L., A novel method for asphalt pavement crack classification based on image processing and machine learning. Engineering with Computers, Vol.35, Issue 2, pp.487-498, 2019.

[6] Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Chen, C., Yang, G., Li, J., Yang, E., and Qiu, S., Automated pixel ‐ level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. Computer ‐ Aided Civil and Infrastructure Engineering, Vol. 34, Issue 3, pp.213-229, 2019.

[7] Nhat-Duc, H., Nguyen, Q. L., and Tran, V. D. , Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. Automation in Construction, Vol. 94, pp.203-213, 2018.

[8] Zhang, D., Li, Q., Chen, Y., Cao, M., He, L., and Zhang, B., An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. Image and Vision Computing, Vol. 57, pp.130-146, 2017.

[9] Zhang, L., Yang, F., Zhang, Y. D., and Zhu, Y. J., Road crack detection using deep convolutional neural network. In 2016 IEEE international conference on image processing (ICIP), pp. 3708-3712, 2016.

[10] Chun, P., Yamane, T., and Tsuzuki, Y., Automatic Detection of Cracks in Asphalt Pavement Using Deep Learning to Overcome Weaknesses in Images and GIS Visualization. Applied Sciences, 11(3), 892, 2021.

[11] Saar, T., and Talvik, O., Automatic asphalt pavement crack detection and classification using neural networks. In 2010 12th Biennial Baltic Electronics Conference, pp. 345-348, 2010.

[12] Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., and Wang, S., Deepcrack: Learning hierarchical convolutional features for crack detection. IEEE Transactions on Image Processing, Vol.28, Issue 3, pp.1498-1512, 2018.

[13] Chun, P., Izumi, S., and Yamane, T., Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. Computer-Aided Civil and Infrastructure Engineering, pp.1-12, 2020.

[14] Park, S., Ahmad, S., Yun, C. B., and Roh, Y. , Multiple crack detection of concrete structures using impedance-based structural health monitoring techniques. Experimental Mechanics, Vol. 46, Issue 5, pp.609-618, 2006.

[15] Sohn, H. G., Lim, Y. M., Yun, K. H., and Kim, G. H., Monitoring crack changes in concrete structures. Computer-Aided Civil and Infrastructure Engineering, Vol. 20, Issue 1, pp.52-61, 2005.

[16] Chun, P., Ujike, I., Mishima, K., Kusumoto, M., and Okazaki, S., Random Forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results. Construction and Building Materials, Vol.253, 119238, 2020.

[17] Okazaki, Y., Okazaki, S., Asamoto, S., and Chun, P., Applicability of machine learning to a crack model in concrete bridges. Computer ‐

Aided Civil and Infrastructure Engineering, 2020.

[18] Chun, P., Yamane, T., Izumi, S., and Kameda, T., Evaluation of tensile performance of steel members by analysis of corroded steel surface using deep learning. Metals, Vol. 9, Issue 12, pp.1259, 2019.

[19] Chun, P., Tsukada, K., Kusumoto, M., and Okubo, K., Investigation and repair plan for abraded steel bridge piers: case study from Japan. Proceedings of the Institution of Civil Engineers-Forensic Engineering, Vol. 172, Issue 1, pp.11-18, 2019.

[20] Chun, P., Yamane, T., Izumi, S., and Kuramoto, N., Development of a Machine Learning-Based Damage Identification Method Using Multi-Point Simultaneous Acceleration Measurement Results. Sensors, Vol. 20, Issue 10, pp.2780, 2020.

[21] Redmon, J., and Farhadi, A., Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[22] Zahid H., Samsoon N., Tasmia R. and Shahnewaz S., Pothole and Speed Breaker Detection Using Smartphone Cameras and Convolutional Neural Networks 2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020, Dhaka, Bangladesh.

[23] Song, W., Jia, G., Zhu, H., Jia, D., and Gao, L., Automated pavement crack damage detection using deep multiscale convolutional features. Mathematical Problems in Engineering Vol. 2020, Article ID 6412562, 2020.

[24] Nhat-Duc H. and Quoc-Lam N., Automatic Recognition of Asphalt Pavement Cracks Based on Image Processing and Machine Learning Approaches: A Comparative Study on Classifier Performance. Mathematical Problems in Engineering, Vol.2018, Article ID 6290498, 2018.

[25] LabelImg, https://github.com/tzutalin/labelImg (accessed Jun. 29, 2020).

[26] Kamal, A.Yolo, yolov2 and yolov3: All you want to know, https://medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899, accessed Jun. 29, 2020), 2019.