Article

Automatic Detection of Cracks in Asphalt Pavement Using Deep Learning to Overcome Weaknesses in Images and GIS Visualization

Pang-jo Chun [1],*, Tatsuro Yamane [2] and Yukino Tsuzuki [3]

[1] Department of Civil Engineering, The University of Tokyo, Tokyo 113–8656, Japan
[2] Department of International Studies, The University of Tokyo, Chiba 277-8561, Japan; yamane.tatsuro.20@dois.k.u-tokyo.ac.jp
[3] Department of Civil and Environmental Engineering, Ehime University, Ehime 790–8577, Japan; tsuzuki.yukino.15@cee.ehime-u.ac.jp
* Correspondence: chun@g.ecc.u-tokyo.ac.jp; Tel.: +81-3-5841-0442

Featured Application: This technology can contribute to improving the efficiency and accuracy of pavement inspection.

Abstract: The crack ratio is one of the indices used to quantitatively evaluate the soundness of asphalt pavement. However, since the inspection of pavement requires much labor and cost, automatic inspection of pavement damage by image analysis is required in order to reduce the burden of such work. In this study, a system was constructed that automatically detects and evaluates cracks from images of pavement using a convolutional neural network, a kind of deep learning. The most novel aspect of this study is that the accuracy was recursively improved through retraining the convolutional neural network (CNN) by collecting images which had previously been incorrectly analyzed. Then, study and implementation were conducted of a system for plotting the results in a GIS. In addition, an experiment was carried out applying this system to images actually taken from an MMS (mobile mapping system), and this confirmed that the system had high crack evaluation performance.

Keywords: deep learning; convolutional neural network; artificial intelligence; pavement; crack; crack detection; GIS

1. Introduction

The increased distress on roadway pavements owing to the heavy loads from vehicles adversely affects the durability of pavement structures, the drivability of road surfaces, and overall road conditions [1,2]. Pavement management systems (PMS) are employed by highway administrations to maintain good driving conditions, and for efficiently manage constant road maintenance and repair works. As a necessary input of the PMS, present pavement condition is included, which provides an evaluation of the road network [3,4]. There are several characteristics that can be assessed in pavements, but are usually classified in surface characteristics (including longitudinal profile, roughness, and surface texture and skid resistance), pavement distresses, structural evaluation, and sub-surface characteristics. However, there is not a universal approach and each highway administration collects pavement condition data following its own criteria [5,6]. Moreover, there are various indices for measuring the same characteristic [7,8].

Several methods have been established for collecting above mentioned information. For example, the falling weight deflectometer (FWD) test is used to assess pavement-bearing capacity [9]; however, since the FWD test needs to be carried out during a road closure, it can only be done in detailed surveys. Alternative assessment methods use the pavement serviceability index (PSI) [10,11] and the maintenance control index (MCI) [12,13]. The PSI is used to identify visible distress on pavement, such as cracking, patching, slope

variance, and rut depth. In Japan, the MCI is used as an index of road pavement conditions instead of the PSI, which is defined by crack ratio, rut depth, and smoothness. However, in order to obtain the PSI and MCI comprehensively, it is necessary to measure the unevenness of the road surface using laser technology or other means, which makes the inspection costs high. In addition, due to the difficulty of procuring the inspection system, such inspections are rarely conducted except for critical roads such as highways. However, even on public roads, it is necessary to avoid the worsening of driving performance and safety due to pavement deterioration, so it is necessary to conduct inspections and make necessary repairs. Therefore, in the inspection of pavements, it is common to evaluate the crack ratio, which is relatively easy to measure. Cracking is one of the most common asphalt pavement road failures and an important indicator of structural deficiencies [14]. The most common crack evaluation method is visual evaluation by engineers during a patrol. However, this method has the disadvantage of being labor-intensive and inconsistent because it involves subjective judgments [15,16]. In addition, the visual inspection requires an enormous amount of recording time and is prone to error. Therefore, there is a need for innovation in road inspection through the development of efficient and quantitative crack evaluation methods.

In recent years, some studies have been carried out on ways to automatically detect cracks using a digital camera mounted on a vehicle and image processing from the images taken. This allows the inspection to be carried out simply while driving the vehicle, which significantly reduces the amount of labor required during the inspection. In addition, because the analysis is performed automatically from the images taken, there are no individual differences among the inspectors, so consistency can be maintained. For example, the Single Shot Multibox Detector (SSD) was used by Maeda et al. [17] and You Only Look Once v2 (YOLOv2) was used by Mandal et al. [18] to detect the location of cracks. However, these object detection methods can only be used to specify the cracks in the image as rectangles, and it is difficult to evaluate the degree of cracking. On the other hand, a method to detect cracks at pixel level is also considered. For example, Kanaeva and Ivanova [19] performed semantic segmentation using U-Net, and Kalfarisi et al. [20] performed semantic segmentation using a faster region-based convolutional neural network (Faster R-CNN) and Mask R-CNN. However, in order to achieve this, it is necessary to specify which pixels are cracks in each pixel in the image of the training data, so the time required for the annotation process is huge. At any rate, as mentioned above, there are many techniques, but since deep learning itself is a new method, research is still in its early stages and the accumulation of technology and development of methods is still required.

In this study, we present a methodology for evaluating the presence of cracks in an image by dividing the image into overlapping segments. The present method is positioned as one that can obtain the crack ratio correctly unlike object detection methods such as SSD, and the effort of annotation is much smaller compared to semantic segmentation techniques such as Mask R-CNN. Based on our experience in [21–26] with research on damage detection using machine learning and deep learning, we believe that to improve the accuracy of deep learning, it is necessary not only to increase the amount of data but also to improve the quality of the training data. Therefore, in this study, we develop a framework to focus on training of types of images that are difficult to detect, from the viewpoint of improving the training performance. Specifically, we built a method to improve accuracy recursively by retraining CNNs by collecting images whose analysis results were wrong. We confirmed that this method is more effective than simply increasing the number of training images. We also proposed an evaluation formula to adapt to the conventional inspection scheme, and constructed a system to plot these formulas on a GIS, which are also described in this paper.

2. Evaluation of Pavement Crack Damage by Deep Learning

2.1. Outline of Convolutional Neural Network

In this paper, we develop a method for detecting cracks using a convolutional neural network (CNN), a type of deep learning. CNN models the receptive field in the human field of vision and is known to have a high level of performance in the field of image recognition. We present the outline of CNN below considering the convenience of readers; more detail regarding the approach can be found in previous studies [27,28]. CNN differs from traditional neural networks in that it has two special layers: a convolutional layer and a pooling layer. A schematic of a typical CNN structure is shown in Figure 1. First, an image is added to the input layer, followed by repeated calculations in the convolutional and pooling layers. The fully connected layer performs weighted connection computations, as is also done in traditional neural networks. In the output layer, the classification results are output. In this section, we first give an overview of CNN for the reader's convenience. Please refer to [28] for details.
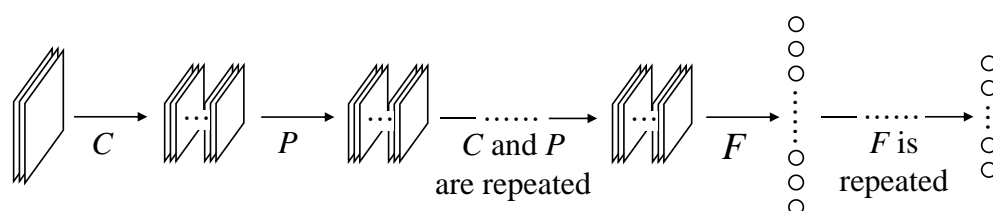

Figure 1. Basic structure of convolutional neural network (CNN) (*C* = convolutional layer, *P* = pooling layer, and *F* = fully connected layer).

2.1.1. The Convolutional Layer

The convolutional layer performs the operation of convolving the filter with respect to the provided input, and thus this layer has the function of identifying local characteristics. The basic structure of the convolutional layer is shown in Figure 2. Assuming that the size of images inputted to the convolutional layer is $W \times H \times K$, the size of the filter is given by $w \times h \times K \times L$. W is the image width, H is the image height, K is the number of channels of the image inputted to the convolutional layer, w is the width of the filter, h is the height of the filter, and L is the number of filters.
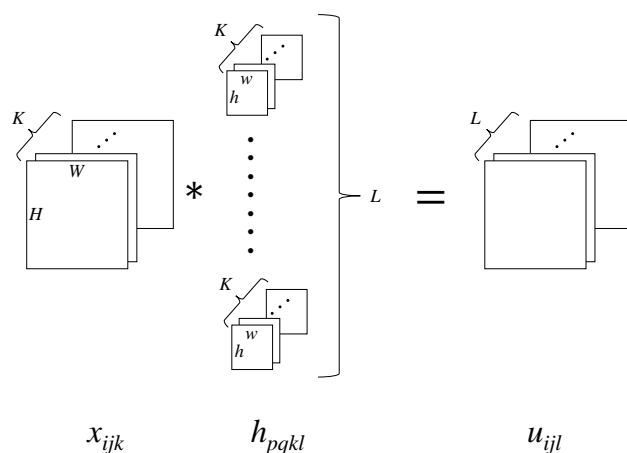

Figure 2. Schematic of the calculation performed in the convolutional layer.

If the pixel value of the input image is expressed using x_{ijk} with the index (i, j) ($i = 0$, ..., $W - 1$, $j = 0$, ..., $H - 1$, $k = 1$, ..., K) and the pixel value of the filter using h_{pqkl}

$(l = 1, \ldots, L)$ with the index (p, q) $(p = 0, \ldots, w - 1, q = 0, \ldots, h - 1)$, the convolution operation is expressed with Equation (1):

$$u_{ijl} = \sum_{k=1}^{K} \sum_{p=0}^{w-1} \sum_{q=0}^{h-1} x_{i+p,j+q,k} h_{pqkl} + b_l \tag{1}$$

where u_{ijl} is the pixel number of the output image and b_l is bias.

Next, we applied the activation function f to u_{ijl} obtained with Equation (1) as shown in Equation (2):

$$z_{ijl} = f(u_{ijl}) \tag{2}$$

While activation functions such as sigmoid function, tanh, normalized first-order function called ReLU, and soft sign function have been proposed so far, ReLU which was reported to be the most suitable was used in this study [29]. ReLU is expressed using Equation (3):

$$f(u) = \max(u, 0) \tag{3}$$

The output image resulting from these analyses will be used as input for the next layer.

2.1.2. The Pooling Layer

The pooling layer is usually placed immediately after the convolutional layer and by lowering the positional sensitivity of the filter response obtained in the convolutional layer, invariance against microscopic horizontal changes can be achieved. The pooling layer obtains a representative value for a part of the group of pixels in the input image and uses this value as the pixel value for the new output image. In image analysis, it is common to consider the maximum pooling that uses the maximum value as the representative value. We used this approach in the present study.

2.1.3. The Fully Connected Layer

The provided input image is one-dimensionally operated in the fully connected layer, and all input and output units are connected. In the final output layer, the effective thickness is output as a continuous value, and learning is performed such that the sum of the squared errors for the output and the target output (output provided as the teacher) would be minimum values. In this manner, an ideal weight can be obtained.

2.1.4. ResNet

In this study, we apply ResNet [30], which is a CNN network model. Since the breakthrough of the CNN in the field of image recognition, its accuracy has been improved by adding deeper layers. However, the deepening of the layers caused a degradation problem. The degradation problem is a phenomenon in which the improvement in training errors in a deep-layered model reaches its peak earlier than in a shallow-layered model. A model with deeper layers should have the same or better training errors than a model with shallower layers, but the accuracy is not easy to be increased and deteriorates rapidly as the depth of the network increases. The difference between ResNet and conventional CNNs is that ResNet learns the residual function with reference to the input of the layers. A part of the general network and ResNet is shown in Figure 3. Consider the case where the function we want to train is (x). In ResNet, in two consecutive convolutional layers, the input x is skipped and connected to the output two layers away. In this case, the difference between the input (x) and the input x is given by Equation (4), and we proceed with the learning based on a modification of this Equation (5).

$$F(x) := H(x) - x \tag{4}$$

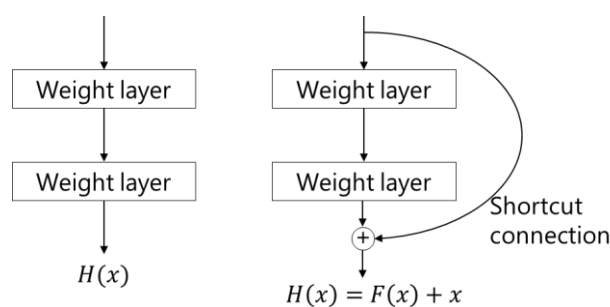$$H(x) := F(x) + x \tag{5}$$

Figure 3. Residual block and shortcut connection (**left**: general network, **right**: residual block).

This replaces the problem of estimating the optimal function H with the problem of estimating the optimal residual function F. The shortcut connection acts as a detour to combine the input values of the layers with the output of the network in front of the activation function, and since the shortcut connection transmits the input information as it is, the gradient also transmits in the back propagation. Therefore, the gradient remains without the risk of being too small or too large. A variety of forms of ResNet have been proposed, such as ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, among others, which are different in the number of layers and the number of parameters that can be learned. ResNet is so powerful that there are several examples of its use in pavement crack detection, as shown in [31,32]. In this study, ResNet50 is used in consideration of the balance between accuracy and computation time.

2.2. CNN Training for Accurate Detection of Cracks in Pavements

2.2.1. Input Images

The size of the original images obtained from the vehicles used in this study is 1024 $\times$ 1024 pixels. In this study, we divide each image into 256 $\times$ 256-pixel sub-regions and give them as inputs for CNN. The method of image division is shown in Figure 4. Firstly, we segmented the 256 $\times$ 256-pixel image on the upper left corner (the red square in Figure 4) for analysis, then moved the segmentation area of 256 $\times$ 256 pixels by 128 pixels to the right (the red square with a dashed line in Figure 4) for the next analysis. When the segmentation process reached the right edge of the photo image, we moved back to the left edge and slid the segmentation area 128 pixels downward (the red square with a dashed line in Figure 4) to continue the process to the right edge for analysis. Segmenting the target image in this manner creates overlapping areas; the same area will go through analysis four times in total except for the corner and edge areas. If the overlapping area is not created in this way, the cracks on the edge of the small divided image are potentially overlooked, so we adopted this overlapping approach in this study.
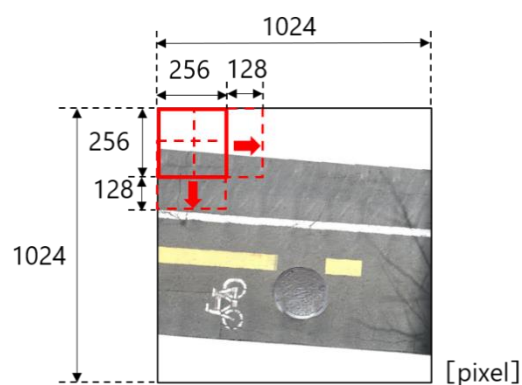

Figure 4. Creation of small segmented images.

2.2.2. Training of the CNN

In this project, we are going to construct a CNN that analyzes small segmented images to judge if the image is a crack or not, as shown in Figure 5. It would be much easier and simpler if we could classify the output with the mere presence/absence of cracks. The result of our initial analysis, however, included false detections in images, where road markings or utility holes were detected as cracks. We, therefore, set six classes as shown in Table 1 to categorize the target images.
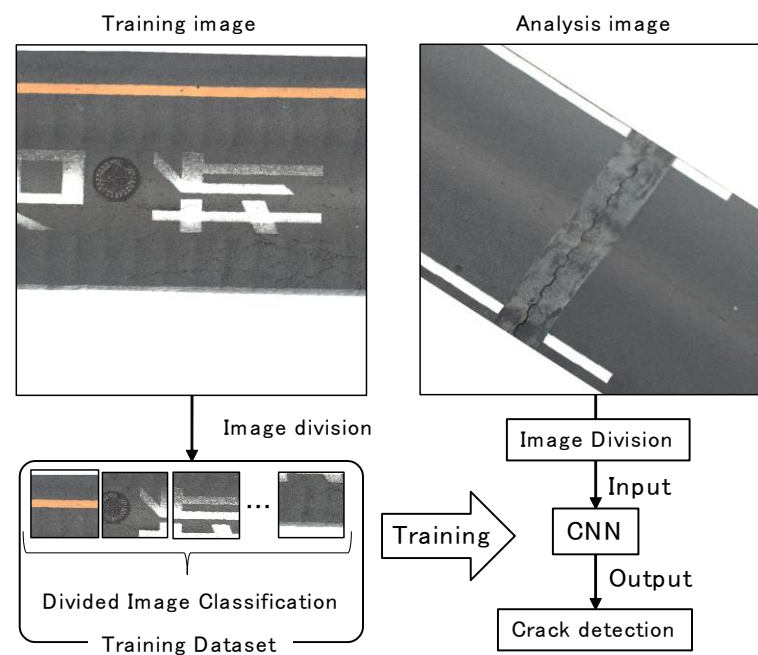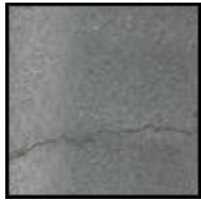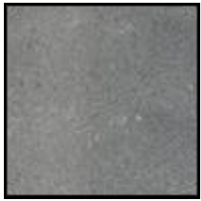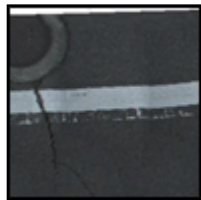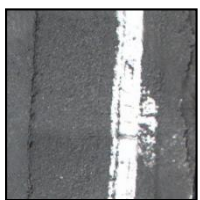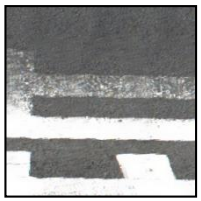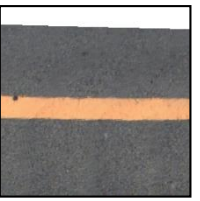

Figure 5. The flow of training and analysis with the CNN in this study.

These six classes of image are used to train the model. In general, the accuracy of deep learning models can be expected to improve as the amount of training data increases. However, simply increasing the amount of training data does not guarantee high accuracy, since the quality of the training data also affects the accuracy. In this study, we tried to develop a framework to improve the accuracy of the model efficiently by increasing the amount of data, selecting data that contributes to the improvement of accuracy. Specifically, we considered that adding images that are likely to be mistaken by the CNN model to the training data would be more efficient than randomly adding training data. The flowchart is shown in Figure 6. First, we perform training on a small randomly selected training data set. This is the initial model. We then let the model classify the new data not used for training (Verification A). Then, after the analysis in verification A is performed, images that could not be properly classified are added to the training dataset. The above flow is considered to be one cycle, and this cycle is repeated. In addition to the data to be added to the training data (the data used for verification A), we also prepare data for calculating the accuracy of the model (verification B). Then, when the accuracy converges, the repetition is stopped. In this study, we decided to stop when the accuracy did not improve for three consecutive cycles.

Table 1. Image classification.

Class Number	Description	Sample Image
1	Cracked area	
2	Non-cracked area	
3	The area including road markings like white lines (with cracks)	
4	The area including road markings like white lines (without cracks)	
5	The area including road facilities like utility holes or bridge joints without white lines (with cracks)	
6	The area including road facilities like utility holes or bridge joints without white lines (without cracks)	

In the case of the images used in this study, it took nine cycles for the accuracy to converge. Table 2 shows the amount of training data in each cycle, the accuracy in verification A and that in verification B. The accuracy was evaluated based on the existence or absence of cracks. For example, mistaking class 3 in Table 1 for class 1 does not count as a mistake. In this study, verification A used 30,000 pieces of data per cycle, and verification B used 20,000 pieces of data per cycle. The data in verification A was then added anew in each cycle. In the training, the ratio of training data to validation data was 4:1, and the number of epochs is 100. The final model is when the validation loss is minimized. In the final model, training loss and validation loss do not diverge, and there is no clear evidence

of overfitting. Note that the images used here and in the following Section 3 were taken in the same city, but show different routes.
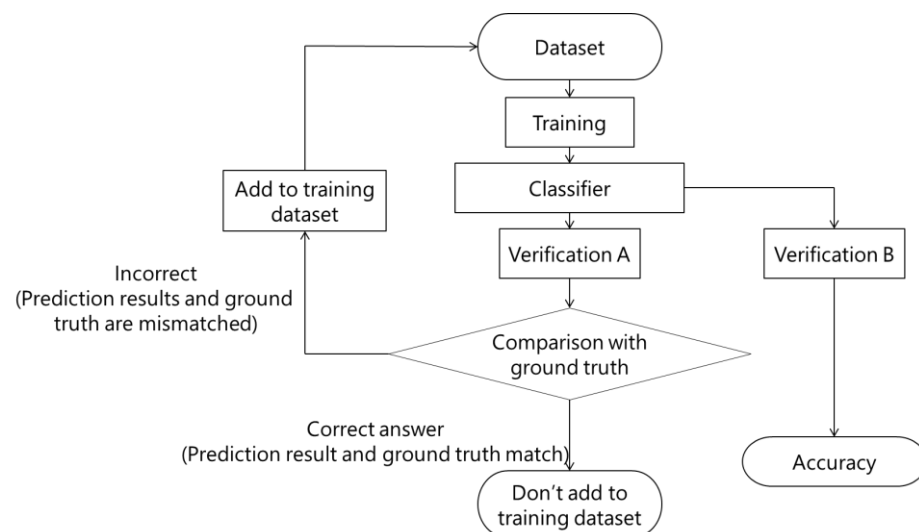

Figure 6. Flow chart for efficient data increase.

Table 2. Training process and improvement of accuracy.

The Number of Cycles	The Number of Images in Dataset	Accuracy in Verification A	Accuracy in Verification B
1	30,000	0.56	0.59
2	43,200	0.74	0.68
3	51,000	0.67	0.65
4	60,900	0.7	0.82
5	69,900	0.85	0.81
6	74,400	0.93	0.96
7	76,500	0.92	0.93
8	78,900	0.97	0.95
9	79,800	0.95	0.94

The improvement of the analysis results is shown in Figure 7. For better understanding of the figure, we do not overlap the images when dividing them, but simply divide them into 256×256 segments. The areas framed in red are those where cracks were correctly detected, the areas framed in yellow are those where cracks were present but not judged to be present, and the areas framed in green are those where cracks were absent but judged to be present. From Step 1 to Step 3, the classifier is confused, probably because of the addition of complex images to the training data, and the accuracy is reduced for this image. However, the accuracy improves again with each step, and finally good results are obtained.

For reference, we also compared to the case of randomly increasing the number of images. The results are shown in Figure 8. It is clear from the figure that although the accuracy is inferior at the beginning, it is superior at the end. This may be due to the fact that the classifier is confused at the initial stage of adding difficult data, but gradually stabilizes. For comparison, Figure 8 also shows the results of training with all the data used in the analysis, not just the data that was incorrect in verification A. Although the number of data is very different from that of the proposed method, the analysis shows that the accuracy is similar or slightly inferior to that of the proposed method. This may be

due to the lower ratio of images in the training dataset that are difficult to analyze. Thus, the proposed method enables efficient learning by selecting images that the classifier is not good at classifying. In many cases, such images included manholes, railroad tracks, drainage ditches, shoulders, bridge joints, automobiles, and shadows.
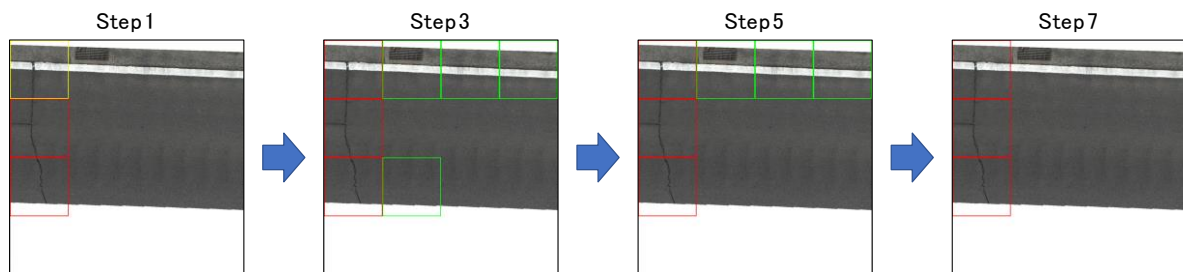

Figure 7. Improvement of the analysis results with an increase in the number of steps.
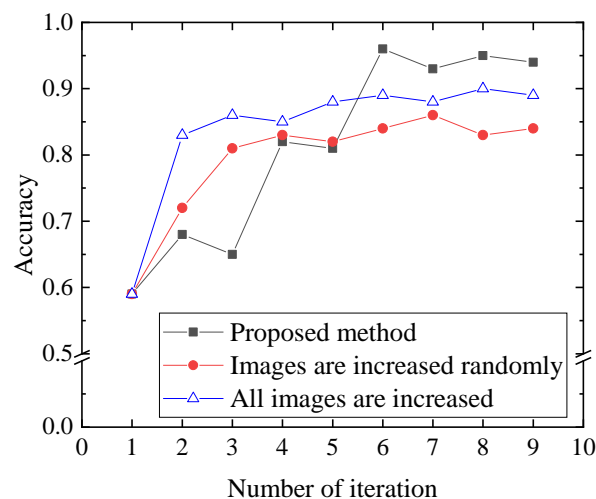
Figure 8. Comparison of accuracy between the proposed and conventional methods.

2.3. Crack Damage Evaluation and Mapping to GIS

Firstly, we defined the percentage value of images with cracks in the segmented target images as *CR* defined by Equation (6).

$$CR = \frac{\sum_{i=1}^{n} p_i}{n} \tag{6}$$

p_i is the ratio of images that contain cracks among the overlapped input images in the pixel of interest (for example, if three of the four overlapping images are judged to be cracked, the value is 0.75), and n is the total number of pixels. The results of the overlap analysis are given, for example, in Figure 9 below. This image is designed to have a pixel value of $p_i \times 255$. That is, it is white for $p_i = 1.0$ and black for $p_i = 0.0$. In Figure 9, a semi-transparent overlay of the original image and the analysis results is shown for clarity.

| Original image | Analysis results | Overlay image |

Figure 9. Improvement of the analysis results with an increase in the number of steps.

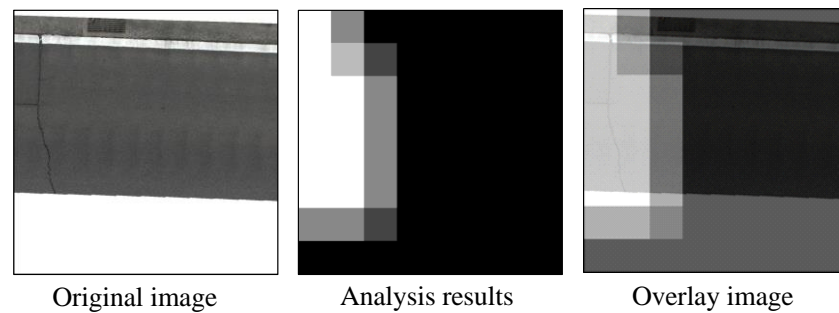Although *CR* is the quantitative value, we should consider the applicability of our method to the existing inspection guidelines. Since the general inspection guidelines stipulate that the degree of crack damage should be evaluated in multiple grades, the thresholds for converting *CR* into multiple grades of evaluation were calculated as follows.

We evaluated the degree of crack damage of the target images first according to the existing inspection guidelines. Then, in order to reproduce the results of the existing inspection guidelines from the *CR*, the mean absolute error of the following Equation (7) is minimized.

$$\text{Mean absolute error} = \frac{1}{N}\sum_{i=1}^{N}|r_{CR} - r_h|_i \tag{7}$$

In the above equation, N represents the number of target images, r_h represents the multiple grade evaluation result under the existing inspection guidelines, and r_{CR} represents the multiple grade evaluation result by *CR*. Specifically, r_{CR} is calculated from *CR* by determining the threshold between grades. Note that the pixels trimmed in the image are also counted in the total number of pixels. This is because if trimmed in the image is excluded from the calculation, some images with a very small denominator will appear, and as a result, even small cracks are often misidentified as serious cracks.

In this study, *CR* are compared with the Japanese pavement inspection procedure, which is judged in three grades (grade 1 to grade 3). Grade 1 means sound, and grade 3 means severely damaged. The threshold values were calculated according to Equation (7) by the data taken in this study; the threshold between grade 1 and grade 2 was 0.33 and the threshold between grade 2 and grade 3 was 0.70. Lastly, we conducted mapping of the resulting data on GIS. In this study, the latitude and longitude information obtained by the vehicle is linked to the acquired images; therefore, it is easy to plot the analysis results on the GIS based on the information.

3. Analysis on Real Roads and GIS Plotting of Evaluation Results

3.1. Photographing and Preparation of Pavement Images

In this section, we analyze images taken on a public road in Ehime Prefecture, Japan, using our proposed method. GMS3, a 3D MMS (mobile mapping system) developed by Canaan Geo Research, was used to photograph the road surface (Figure 10). It is possible to photograph a maximum road width of two meters while driving at a maximum speed of 80 km/h. The captured image was saved in GeoTIFF format (1024 × 1024 pixel). The outside of the road was trimmed and removed in pre-processing. The trimmed image can be seen in Figures 4, 5 and 7.

Figure 10. Road Photography with GMS3.

3.2. Pavement Damage Evaluation and GIS plotting

First, the model developed in Section 2.2 (Step 9) was used to detect cracks. The results are shown in Figure 11. Two results are shown for each of the six images. In the images on the left, the areas framed in red are those where cracks were correctly detected, while the areas framed in yellow are those where cracks were present but not judged to be present, and the areas framed in green are those where cracks were absent but judged to be present. For better understanding of the figure, we do not overlap the images when dividing them, but simply divide them into 256 × 256 segments as in Figure 7. The figure on the right shows the results of the analysis as shown in the middle figure of Figure 9. Note that the analysis was performed with overlapping images, as explained in Figure 4.

The results A and B are examples of properly detected cracks; the results A and B have different brightness levels, but are properly detected. Result C is a case in which there was a false positive detection, but in reality the classification is not necessarily wrong, because the image is not easy to determine whether there are thin cracks around the manhole or not. In other words, even the inspector is not sure whether it is correct to conclude that there are cracks or not around the manhole. In the six-class classification shown in Table 1, the classifier judges that the image is classified as class 5. In other words, the classifier recognizes that there is both a manhole and a thin crack in the image. Result D was concluded by the classifier to have cracks in the white area, even though there were no cracks. It is easy to mistakenly detect cracks where there are white lines. Result E is an example where the classifier misidentified a thin shadow as a crack. These thin shadows can easily be mistaken as cracks because cracks are characterized by their dark and thin nature. Result F is an example of overlooking cracks that are overlapping shadows, in addition to false positives like Result E. In order to reduce such errors, it would be necessary to train the classifier with more cases like these, but we could not do so because there were not many of them in the data we collected. Nevertheless, the overall accuracy of the crack detection was very high and was sufficient for the purpose of pavement inspection.

Then, the crack detection results obtained here were evaluated and assigned three grades according to Equation (6). The results are shown in Table 3. We obtained 94.3% accuracy, which indicates that there is a certain degree of agreement between visual inspection and model judgment. In addition, the model showed that less than 1% (5/601) of grade 3 cases were misjudged as grade 1, indicating that the probability of overlooking significant damage is very small.
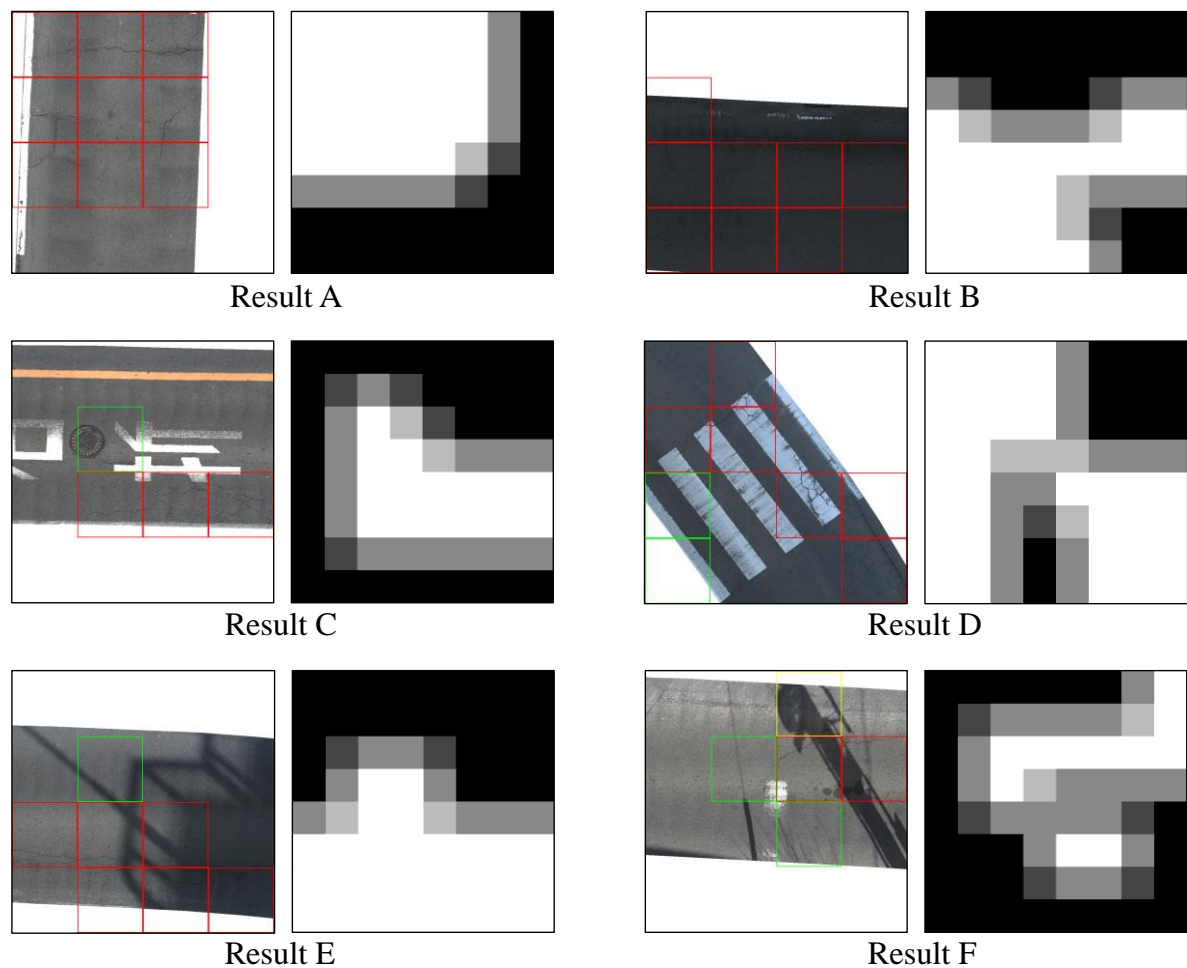
Figure 11. Crack detection results.

Table 3. A confusion matrix with three grades.

Analysis Result Visual Inspection	Grade 1	Grade 2	Grade 3
Grade 1	47,058	2835	132
Grade 2	309	7773	25
Grade 3	5	34	562

The images we used in the project are linked to the longitude/latitude and easily visualized on GIS. Figure 12 shows the results of visual inspection obtained by following the Japanese pavement inspection procedure and evaluation by our method on GIS. A wide area view and an enlarged view are shown. As shown in the figure, our method replicated the visual inspection results with satisfactory accuracy. In particular, if we look at the enlarged view, we can see that the damage is more frequent at intersections, and this may be due to the large external forces acting on them, such as braking. As shown above, the system proposed in this paper, which automates the plotting of pavement conditions on a GIS, is accurate and can contribute to tremendous labor saving in pavement inspection.
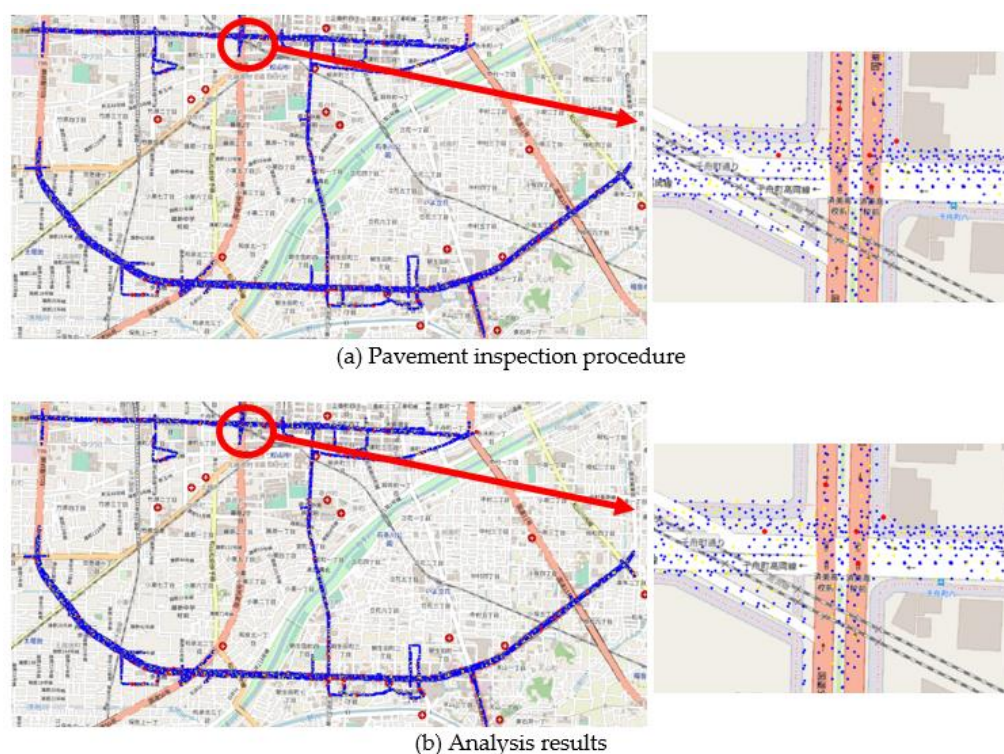
(a) Pavement inspection procedure

(b) Analysis results

Figure 12. Comparison of pavement inspection procedure and analysis results on GIS.

4. Conclusions

In this study, a system to efficiently increase the accuracy of the classifier in the automatic detection of cracks on asphalt pavement surfaces using a deep learning model was constructed and validated. Compared to a classifier trained with randomly selected data, the accuracy of the classifier was improved when intentionally selected data (in this case, incorrect images from the previous step) were used for training as in Table 2 and Figure 8. The method proposed in this study to train mainly using incorrect images from the previous step can eliminate the possibility of reducing the overall accuracy by fitting too much to images that classifier can detect cracks easily. In addition, a method was proposed for evaluating the pavement condition based on the results of crack detection by the model and mapping the results to GIS following existing inspection procedures.

Future issues to be considered are as follows. In the interpretation of Equation (7), trimmed pixels are also included in the calculation, but it may cause underestimation of damage. It is considered that better damage evaluation can be achieved by refining the handling of trimmed pixels. In addition, the model developed in this study showed high detection accuracy for pavement images with cracks, but there were a few false positives in pavement images without cracks. Of particular importance, some false positives were found, caused by slender shadows and joints. In the future, we aim to improve the accuracy by supplementing the training data with more images similar to the false positive images.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, resources, writing—review and editing, supervision, P.C.; methodology, investigation, T.Y.; methodology, investigation, data curation, writing—original draft preparation, visualization, Y.T. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Matsuyama City, Japan and are available with the permission of them.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pais, J.C.; Figueiras, H.; Pereira, P.; Kaloush, K. The pavements cost due to traffic overloads. *Int. J. Pavement Eng.* **2019**, *20*, 1463–1473. [CrossRef]
2. Xiao, D.X.; Wu, Z. Using systematic indices to relate traffic load spectra to pavement performance. *Int. J. Pavement Res. Technol.* **2016**, *9*, 302–312. [CrossRef]
3. AASHTO. *Pavement Management Guide*, 2nd ed.; American Association of State Highway and Transportation Officials (AASHTO): Washington, DC, USA, 2012.
4. Pérez-Acebo, H.; Linares-Unamunzaga, A.; Abejón, R.; Rojí, E. Research trends in pavement management during the first years of the 21st century: A bibliometric analysis during the 2000–2013 period. *Appl. Sci.* **2018**, *8*, 1041. [CrossRef]
5. Zeiada, W.; Hamad, K.; Omar, M.; Underwood, B.S.; Khalil, M.A.; Karzad, A.S. Investigation and modelling of asphalt pavement performance in cold regions. *Int. J. Pavement Eng.* **2019**, *20*, 986–997. [CrossRef]
6. Pérez-Acebo, H.; Gonzalo-Orden, H.; Findley, D.J.; Rojí, E. Modeling the international roughness index performance on semi-rigid pavements in single carriageway roads. *Constr. Build. Mater.* **2020**, 121665. [CrossRef]
7. Kırbaş, U. IRI sensitivity to the influence of surface distress on flexible pavements. *Coatings* **2018**, *8*, 271. [CrossRef]
8. Pantuso, A.; Loprencipe, G.; Bonin, G.; Teltayev, B.B. Analysis of pavement condition survey data for effective implementation of a network level pavement management program for Kazakhstan. *Sustainability* **2019**, *11*, 901. [CrossRef]
9. Schmalzer, P.N. *LTPP Manual for Falling Weight Deflectometer Measurements*; Version 4.1; FHWA, US Department of Transportation: Washington, DC, USA, 2006.
10. Hall, K.; Muñoz, C. Estimation of present serviceability index from international roughness index. *Transp. Res. Rec.* **1999**, *1655*, 93–99. [CrossRef]
11. Fuentes, L.; Camargo, R.; Arellana, J.; Velosa, C.; Martinez, G. Modelling pavement serviceability of urban roads using deterministic and probabilistic approaches. *Int. J. Pavement Eng.* **2021**, *22*, 77–86. [CrossRef]
12. Inoue, T.; Kawamura, A. Monitoring and evaluation of road pavement in Japan. In Proceedings of the ITEC Conference, Seoul, Korea, 18–20 September 2001; p. 4.
13. Yashima, A.; Sawada, K.; Murata, Y.; Kariya, K.; Takeda, K.; Kato, I.; Endo, T. Development of automation technique for check and diagnosis of pavement and earthfill structure. *Jpn. Geotech. Soc. Spec. Publ.* **2017**, *5*, 56–61. [CrossRef]
14. Li, S.; Cao, Y.; Cai, H. Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model. *J. Comput. Civ. Eng.* **2017**, *31*, 04017045. [CrossRef]
15. Hoang, N.D.; Nguyen, Q.L.; Tien Bui, D. Image processing–based classification of asphalt pavement cracks using support vector machine optimized by artificial bee colony. *J. Comput. Civ. Eng.* **2018**, *32*, 04018037. [CrossRef]
16. Wang, S.; Tang, W. Pavement crack segmentation algorithm based on local optimal threshold of cracks density distribution. In *International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 298–302.
17. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road damage detection using deep neural networks with images captured through a smartphone. *arXiv* **2018**, arXiv:1801.09454.
18. Mandal, V.; Uong, L.; Adu-Gyamfi, Y. Automated road crack detection using deep convolutional neural networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5212–5215.
19. Kanaeva, I.A.; Ivanova, Y.A. Road Pavement Crack Detection Using Deep Learning with Synthetic Data. In Proceedings of the 14th International Forum on Strategic Technology (IFOST-2019), Tomsk, Russia, 14–17 October 2019; pp. 320–325.
20. Kalfarisi, R.; Wu, Z.Y.; Soh, K. Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization. *J. Comput. Civ. Eng.* **2020**, *34*, 04020010. [CrossRef]
21. Yamane, T.; Chun, P. Crack Detection from a Concrete Surface Image Based on Semantic Segmentation Using Deep Learning. *J. Adv. Concr. Technol.* **2020**, *18*, 493–504. [CrossRef]
22. Chun, P.; Izumi, S.; Yamane, T. Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Comput. Aided Civ. Infrastruct. Eng.* **2021**, *36*, 61–72. [CrossRef]
23. Chun, P.; Yamane, T.; Izumi, S.; Kameda, T. Evaluation of tensile performance of steel members by analysis of corroded steel surface using deep learning. *Metals* **2019**, *9*, 1259. [CrossRef]
24. Chun, P.; Ujike, I.; Mishima, K.; Kusumoto, M.; Okazaki, S. Random Forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results. *Constr. Build. Mater.* **2020**, *253*, 119238. [CrossRef]
25. Karina, C.N.; Chun, P.; Okubo, K. Tensile strength prediction of corroded steel plates by using machine learning approach. *Steel Compos. Struct.* **2017**, *24*, 635–641.
26. Chun, P.; Yamane, T.; Izumi, S.; Kuramoto, N. Development of a Machine Learning-Based Damage Identification Method Using Multi-Point Simultaneous Acceleration Measurement Results. *Sensors* **2020**, *20*, 2780. [CrossRef]
27. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
28. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

30. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261.

31. Ranjbar, S.; Nejad, F.M.; Zakeri, H. An image-based system for pavement crack evaluation using transfer learning and wavelet transform. *Int. J. Pavement Res. Technol.* **2020**, 1–13. [CrossRef]

32. Liu, J.; Yang, X.; Lau, S.; Wang, X.; Luo, S.; Lee, V.C.S.; Ding, L. Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 1291–1305. [CrossRef]