Quantitative Road Crack Evaluation by a U-Net Architecture using Smartphone Images and Lidar Data

Takahiro Yamaguchi[1] | Tsukasa Mizutani[1]

[1]Institute of Industrial Science,
the University of Tokyo

Correspondence
Takahiro Yamaguchi, Institute of Industrial Science, the University of Tokyo,
Komaba 4-6-1, Meguro-ku, Tokyo, Japan.
Email: tyamag@iis.u-tokyo.ac.jp

ABSTRACT

Road cracks are an important concern of administrators. Visual inspection is labor-intensive and subjective, while previous algorithms detecting cracks from optical camera images were not accurate. Furthermore, the actual length and thicknesses of a crack cannot be estimated only from images. Light Detection and Ranging (Lidar) is a standard feature introduced on the latest smartphones. In this research, for completely automatic, accurate and quantitative road crack evaluation using smartphones, an up-to-date segmentation technique, U-Net with morphology transform adopting data augmentation was proposed. Lidar 3D point cloud data of smartphones is linked to color data obtained from cameras. By registering images to Lidar data, geometrical relationships were estimated to calculate the length and thicknesses. The proposed algorithm was validated by a standard database of road cracks and dataset constructed by the authors, showing 95% length accuracy and 0.98 coefficient of determination for thickness estimation irrespective of various crack shapes and asphalt pavement color patterns.

1 INTRODUCTION

Road distresses, especially cracks are important concerns of administrators. The conditions of road surfaces need to be periodically monitored to ensure the safety of road users. When small cracks advance, they are widened and elongated to connect with each other, causing potholes. There are enormous amounts of road infrastructure stocks worldwide (American Society of Civil Engineers, ASCE [2020], Ministry of Land, Infrastructure, Transport and Tourism, MLIT [2021]). In Japan, for example, we have 1,300,000 km distance road surfaces to be maintained. It is almost impossible to evaluate the whole surface by visual inspections, necessitating the development of automatic road crack evaluation methods.

Considering the standards of road surface condition monitoring, the geometries of crack patterns, e.g. lengths and bifurcations are important. For example, in the Japanese standard, after dividing road surfaces into small areas, the numbers of cracks and bifurcations are evaluated for each panel from sketches to calculate a crack ratio (MLIT [2017]). Cracks need to be properly segmented. Lengths should be quantitatively evaluated. In previous research, it is suggested the thicknesses of asphalt pavement cracks are related to the conditions of pavements (Nakamura *et al.* [2020]). In terms of concrete structures, the thicknesses of vertical cracks are related to the extent of corrosion of reinforcement steel rebars (Vidal *et al.* [2004]). It is helpful to estimate the thicknesses of cracks by the algorithm.

It is a common sense of practitioners that cracks are detected by optical camera images. Our and other research groups are utilizing a Mobile Mapping System (MMS) to detect cracks by line-sensor cameras, and potholes and ruttings by Light Detection and Ranging (Lidar) data using laser sensors (Mizutani *et al.* [2022]). Only targeting larger cracks, or by increasing the resolution of laser data, i.e. light section method may detect cracks though it needs special instruments and measurement time for the reconstruction of detailed three-dimensional surface geometries (Guan *et al.* [2015], Jarvis [1983]). Typical measurement systems of radar and infrared thermography are not targeted for vertical cracks (Yamaguchi *et al.* [2019], Cotič *et al.* [2015]). Previous research works about crack detection by camera images are studied in detail in the following section.

In this research, smartphones were utilized to establish a convenient measurement method (Apple inc. [2020], 3D Scanner App [2022]). Figure 1 shows the measurement

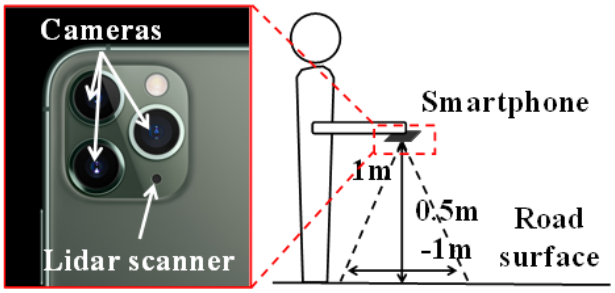
configuration. The method has three advantages: Vehicle-installed type systems are targeted for roadways. Smartphones can be used for sideways and private properties. Administrators are often managing cloud computing systems. Inspectors record road surface conditions by terminals. The system can facilitate inspection processes. Road users can report damages by their smartphones, constructing big data of spatially and temporally dense monitoring results.

2 PREVIOUS RESEARCH

Detecting cracks from images is one of the benchmark tasks of the application of Machine Learning (ML) and Deep Learning (DL) to infrastructure engineering problems. There exist many previous research works. Mainly two targets were discussed: cracks on concrete structure surfaces and asphalt pavement road surfaces. Cracks on concrete surfaces are relatively thin and have complicated structures (Chun *et al.* [2021], Dung & Anh [2019], Fujita & Hamamoto [2011], Kim & Cho [2019], Liu & Gao [2022], Nishikawa *et al.* [2012]). Usually, thickness is smaller than 1 mm. Well-developed cracks have fractal structures. However, they do not necessarily mean road crack detection is an easier task. There are a variety of pavement surface color patterns. Road surfaces may be stained and obstacled by small falling objects with different lighting conditions. Detecting ambiguous cracks from noisy images is not an easy task. However, the detection algorithms for concrete surfaces and road cracks are theoretically the same. Hereafter, we will focus on the previous research about road crack detection. Our method is not targeted but expected to be applicable to concrete and other special types of pavements following the same methodology.

In terms of asphalt pavement road cracks, detection algorithms are classified into three categories: image processing, ML and DL-based methods. Image processing is classical spatial filtering and feature extraction techniques. Zalama *et al.* (2014) adopted Gabor filters to characterize and detect crack patterns on asphalt road images. Zou *et al.* (2012) combined tensor voting, random sampling and graph models to define crack features. Chun & Hashimoto (2015) proposed a semi-automatic crack detection algorithm. The idea is adjusting the crack probabilities of automatic detection results by heuristics developing interactive user interfaces showing updated crack detection results. These methods offered insights into the characteristics of crack geometries. However, the problem of how to set the thresholds of detecting cracks remains. ML and DL-based automatic methods are prevailing to deal with various crack geometries and photographing conditions leveraging the advantages of large databases and advanced computational environments.

Novel ML techniques are being proposed day by day (Bishop [2006]). There are as many ML-based crack detection algorithms as ML techniques. Here the summaries of several important research works are introduced. Shi *et al.* (2016) proposed a crack detection algorithm based on random structured forests. Geometrical patterns of cracks were registered in random forests and judged by a Support Vector Machine (SVM) classification algorithm. A CrackForest segmentation database (CFD) was developed to validate the algorithm. Hoang & Nguyen (2019) combined image filters to extract features of cracks. SVM, Neural Network (NN), and RF were compared to accurately classify crack features. Prasanna *et al.* (2016) detected cracks on concrete bridge decks using automated robots by RF of defined features. The drawback of ML-based methods is effective feature values should be devised to conduct accurate detection. These research works claimed the accuracy of ML is comparable with DL. In this research, we assumed the existence of large-scale crack databases. Convolutional Neural Networks (CNN) of DL can automatically and accurately extract spatial features from databases.



(a) (b)

FIGURE 1. Road crack evaluation by smartphones (Apple inc. [2020], 3D Scanner App [2022]). (a) Latest version smartphones (e.g. iPhone 12 Pro) equipped with cameras and a Lidar scanner. (b) Typical arrangement of the measurement, about 1m height from the road surface covering a 0.5m - 1m square area. Precise settings are not needed. Instead, Lidar data is utilized.

DL-based methods are further classified into three categories depending on a purpose: image classification, localization and segmentation (Goodfellow *et al.* [2016]). Classification is to categorize images into crack or non-crack. Maguire *et al.* (2017) developed image databases of various cracked structures, i.e. concrete bridge decks, walls, and pavements named SDNET2018. Because the locations of cracks should be estimated on the whole road surface image, the trend is localization and segmentation. A simple idea is sliding a target window to crop and classify each region image. To estimate target regions efficiently and adaptively, localization by a region proposal scheme was developed (Girshick [2015], Liu *et al.* [2016], Ren *et al.* [2017], Redmon *et al.* [2016]). Maeda *et al.* (2018), Maeda *et al.* (2021) developed a road distress database with annotation results of regions and damage types from images obtained by a smartphone camera fixed on a vehicle dashboard, called RDD2020. Single Shot MultiBox Detector (SSD) was utilized to localize and categorize damages.

Segmentation is pixelwise localization of cracks and the goal of this research. To accomplish complete quantification of cracks, i.e. estimation of length and thicknesses, segmentation is indispensable. Huang *et al.* (2022) proposed a segmentation database called NHA12D, which is composed of

TABLE 1. Summary of opensource road crack databases with different pavement, view, annotation types with numbers of images.

Database	Target pavement	View	Annotation	Total image No.
SDNET2018 (Maguire et al. (2017))	Concrete (there also exist concrete bridge deck and wall versions)	Top	Classification	2,600 crack and 21,700 non-crack (104 pavements)
RDD2020 (Maeda et al. (2018))	Asphalt (including potholes and other types of road damages)	Quarter	Localization	13,400
CrackForest (CFD) (Shi et al. (2016))	Asphalt	Top	Segmentation	120
NHA12D (Huang et al. (2022))	Asphalt and concrete	Top and Quarter	Segmentation	80 (small but size of each image is relatively large)
Crack500 (Yang et al. (2020), adopted)	**Asphalt**	**Top**	**Segmentation**	**500**

images obtained by cameras installed on a vehicle with top and forwarding views and segmentation teacher data to objectively compare the previous research. Ji *et al.* (2020) segmented asphalt road cracks by a CNN encoder-decoder architecture. Bang *et al.* (2019) proposed a CNN encoder-decoder architecture to segment cracks in black-box images. Yang *et al.* (2020) developed the largest segmentation database called Crack500. To improve segmentation accuracy considering the context of crack images, top-down and bottom-up spatial feature pyramids were combined to propose an elaborated CNN architecture.

The results shown by Yang *et al.* (2020) stated above were the latest and baseline of our research. The problem of the algorithm is that, however, accuracy may be further improved even though it considered the complicated feature pyramid. This research utilizes the knowledge of biomedical engineering, U-Net (Falk *et al.* [2019], Zhou *et al.* [2018]). By incorporating skip structures to encoder-decoder architectures, multi-scale features are naturally introduced to conduct an accurate and low calculation cost efficient segmentation. This research is also related to Huyan *et al.* (2020), which proposed a U-Net structure to segment pavement cracks, though not validated by open databases. The proposal of this research is that accuracy was further improved by data augmentation, and morphology transform (Haralick *et al.* [1987], Peters [1995]) after U-Net segmentation (idea #1). To conduct a fair comparison, an opensource crack database was utilized as training and test data to demonstrate the effect of the algorithm. Discussed databases were summarized in Table 1. Considering the number of segmented images and compared algorithm, Crack500 was adopted. Crack detection from quarter view (forwarded, inclined) images, and concrete and other types of pavements are next steps considered as future works.

The last but not least problem of the previous research is the actual scales of cracks are not known only from images. Furthermore, the resolution is not uniform in an image considering the center and side portions, and slight inclination of a camera. This point is not discussed in previous research. On the other hand, because of the advancement of smartphone sensors, Light detection and ranging (Lidar) is installed as a standard function of iPhone and iPad as shown in Figure 1. Lidar 3D point cloud data is linked to color data obtained from cameras for the purpose of virtual space modelling. Cracks are not wide enough to be detected by Lidar data. The proposal of this research is, comparing images and Lidar data, only by one shot of a smartphone, both the shapes and scales of cracks were understood to estimate the lengths and thicknesses of cracks automatically and accurately, which are the aim of the research (idea #2).

3 PROPOSED ALGORITHM

3.1 Contributions

In response to the two ideas discussed above, two contributions of the research are summarized below.

1) U-Net, the state-of-the-art deep CNN architecture was utilized to accomplish the road crack segmentation task. Data augmentation was adopted. Morphology transform after segmentation was found to show the highest segmentation accuracy compared with the previous research trained and validated by the same Crack500 database.

2) The segmentation results were quantified by registering images to lidar data. Similarity transform was assumed. The images were resampled with known distances to quantitatively estimate the lengths and thicknesses of cracks. The quantification accuracy of these indices was validated by the

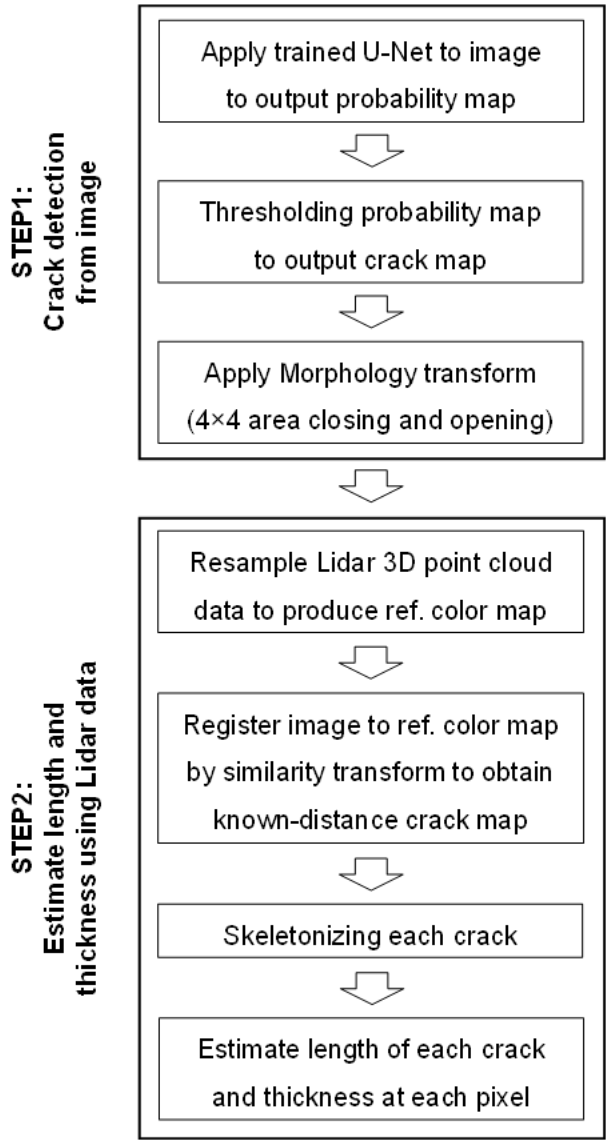


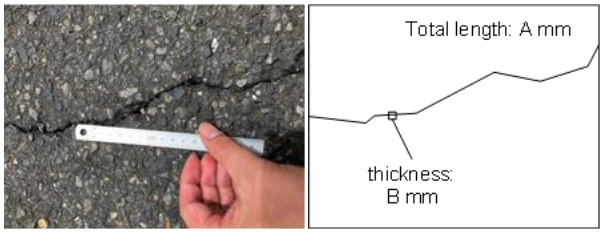
FIGURE 2. Flow chart of the proposed algorithm. The algorithm consists of two steps: crack segmentation by U-Net and quantitative evaluation by Lidar data.

crack data measured by the authors in real situations with various crack shapes and asphalt pavement color patterns.

In the following section the flow chart of the proposed algorithm is explained in detail. The two accuracy-related terms "segmentation accuracy" appeared in contribution #1 and "quantification accuracy" in contribution #2 (Popovic *et al.* [2007], Harada [2017]). The configuration of the data and evaluation methods were defined in the following section 3.3 and section 3.4.

3.2 Outline of the algorithm

Figure 2 explains the flow chart of the proposed algorithm. The algorithm consists of 2 steps. After obtaining an image and Lidar data of a target region by a smartphone, cracks are detected by pixel-level, i.e. segmented (step 1). Utilizing Lidar data, the lengths and thicknesses of detected cracks were quantitatively estimated (step 2). The details of each step are explained in order in the following chapters. The important points are optimizing and training U-Net architectures, application of Morphology transform, matching the image and Lidar data, estimating lengths and thicknesses. In each step, the state-of-the-art image processing techniques and novel ideas were combined to output up-to-date crack segmentation and geometry estimation results.



(a) **(b)**

FIGURE 3. Measuring the geometries of cracks. (a) Lengths and thicknesses by rulers. (b) Sketches.

3.3 Data configuration

In this research, two types of data were used to develop and validate the algorithm. One is Crack500, a crack database explained in the previous chapter to train and validate the proposed U-Net architecture. The details of this database are explained in Yang et al. (2020). The cracks on asphalt pavements in the university campus were photographed by cell phones. The database includes 250 training images, 50 validation images, 200 test images, and the same number of segmentation teacher data for each image. Validation data was used for checking training processes and test data for evaluating the generalization error i.e. segmentation accuracy for unforeseen data. The sizes of most images are about 1500 pixels by 2500 pixels. There are large variations in pavement color patterns, lighting conditions, crack patterns, and thicknesses such as forked and alligator type cracks. However, the effects of shadows, road structures such as manhole, joint, and drainage covers, boundaries of repair works and crack sealings were not considered in both previous and our research. In some cases, they were falsely detected. It is not clear the inclusion of these features in training data will remove false detections or additional analysis is needed, which remains as future works.

The other data is prepared by our group. The applicability of U-Net was confirmed by Crack500. Our data is to validate our quantification algorithm by Lidar data. We searched asphalt pavement surfaces in the campus of Institute of Industrial Science (IIS), the University of Tokyo, Tokyo, Japan and picked up four model crack cases (Figure 3). The images of the cracks were obtained by iPad Pro 11 inch (2nd generation). At the same time, Lidar data was obtained by a commercial software, 3D Scanner App. iPad was placed above the target region to obtain top view images as shown in Figure 1 b. Height was around 1m and covered areas were about 0.5m

- 1m by 0.5m - 1m, though our method is not sensitive to the height and slight inclination of the cameras. The APP also offers the estimated position of an observer, which indicates that 10cm to 20cm variations of the heights of the sensors were not important. Smartphones can measure a horizontal level, which indicates 10 degree to 20 degree inclination variations. iPhone can also be used as measurement devices following the same procedure. Crack #1 to Crack #4 have different crack shapes, thicknesses, pavement color patterns, and obstacles. The details are discussed in chapter 5. The lengths and thicknesses were measured by rulers on site to draw sketches (Figure 3). The thicknesses were relatively large, not requiring crack scales. Because of the complicated crack shapes and fluctuations of thicknesses along the cracks, the error was at most 5mm for length and 1mm for thickness measurement. This point is also discussed in chapter 5.

3.4 Evaluation methods

The two evaluation indices, "segmentation accuracy" and "quantification accuracy" are defined. Segmentation accuracy is to evaluate the segmentation performance of trained U-Net. Segmentation accuracy is the averaged F-measure (dice coefficient) on the whole dataset by adopting the best threshold probability for each image (OIS), following the previous research (Yang *et al.* [2020]). The number of crack pixels is much smaller than background pixels. Therefore, precision, true crack pixels among all the detected pixels P, and recall, detected pixels among all the true crack pixels R are considered. F-measure F is defined below.

$$F = 2 \times \left(\frac{P \times R}{P + R}\right) \quad (1)$$

Deep learning models output the probability of a crack for each pixel. The threshold of crack probability t is the parameter and P, R, F are the function of t. Too large t will judge all the pixels are non-crack and decrease R while too small t will judge all the pixels are crack and decrease P. The best t which provides largest F is estimated for each image (OIS), then averaging OIS of all the images will return segmentation accuracy. This index is compared to demonstrate the efficiency of our proposed method.

Quantification accuracy is to evaluate our algorithm of estimating the lengths and thicknesses of cracks. Our constructed crack dataset contains the information of the total length of cracks in an image and measured thicknesses at the three points of the cracks per one image. The error rate of length was evaluated for four crack images and coefficient of determination of twelve thicknesses plotting estimated versus true thickness values on a scatter plot.

Calculation time was also evaluated to demonstrate the efficiency of our U-Net segmentation model. Programs were written in TensorFlow (TensorFlow [2022]). Training and testing computer environment is GPU: NVIDIA GeForce GTX 1080 ti and CPU: Intel Core i7-8700K @ 3.7GHz (NVIDIA, [2022]). Because the environment is different from the previous research, NVIDIA GeForce GTX TITAN X, a fair comparison is not possible. However, the performance of GTX 1080 ti is close to GTX TITAN X by benchmark tests (PassMark Software [2022]). As will be discussed in the following chapter, it can be inferred that our result is consistent and proposed method is fast. This fact is derived from the simple architecture of U-Net.

4 UP-TO-DATE CRACK SEGMENTATION BY U-NET, DATA AUGMENTATION AND MORPHOLOGY TRANSFORM

In this chapter, a U-Net architecture was optimized. Data augmentation techniques were introduced, and morphology transform was applied to segmentation results to improve segmentation accuracy. The comparison of U-Net and previous architectures, and effects of data augmentation and morphology transform on segmentation accuracy were summarized in section 4.4.

4.1 U-Net architecture

Figure 4 explains the proposed U-Net architecture. The name of U-Net comes from its unique U-shaped architecture. Inputs are RGB crack images, integers from 0 to 255 (uint8) with three channels. Images are batch-normalized in the training process to reduce the effect of brightness differences. U-Net is the cascade of convolution layers. Small 3 by 3 spatial filters with 15 layers were adopted to extract the features of the input images. Increasing the number of layers results in considering more abstract features in larger scale contexts. Larger size filters and shallower layer architectures can be adopted though they do not fit in the concept of U-Net. By increasing the number of filters, large size images are converted into a large number of small feature maps. In segmentation tasks, from extracted features, segmented images should be reconstructed. The former half of the structure is called encoder and latter half is decoder. A special upsampling method, transpose filter was utilized in fully convolutional segmentation networks. The mechanism was explained in the previous research (Falk *et al.* [2019], Zhou *et al.* [2018]). The architecture of the decoder is symmetrical with the encoder. In U-Net, upsampled feature maps are concatenated with encoder feature maps to consider various scale features with different levels of abstraction simultaneously, called skip connections. Integrating all the corresponding layers constructs U-shaped architectures. The strides and sizes of max pooling and transpose filters are 2 by 2 holding the same feature map sizes. As activation functions, rectified linear unit (Relu) functions were applied after each convolution layer and a sigmoid function was utilized to the last convolution layer to output probabilities. The final output is the probability of a crack for each pixel. All the training parameters were initialized following normal distributions.

The important hyperparameter of U-Net is the number of layers. The effect of the number of layers was evaluated by comparing a) the proposed 15-layer model, b) removing the last three convolution layers of the encoder and first three ones of the decoder (9-layer model), c) adding three convolution layers after the last layer of the encoder and three ones before the first layer of the decoder (21-layer). The effect of skip connections can be evaluated by comparing the performances

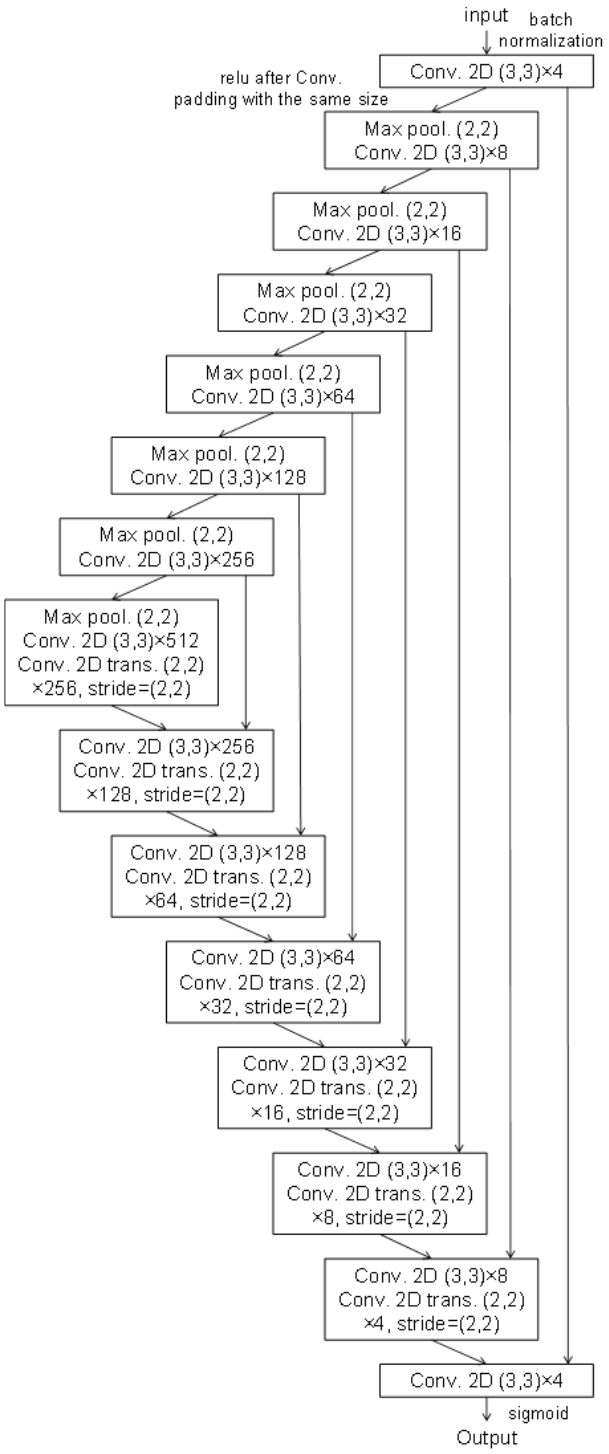


FIGURE 4. Proposed U-Net architecture. Characteristics of the architecture are symmetric encoder-decoder structure and skip connections.

of the proposed and previous architectures. The results were discussed in section 4.4.

Important training parameters were a loss function, optimization method, learning rate, batch size and number of epochs. The difficulty of this task was that the portions of cracks were much smaller than background areas, which caused the failure in a training process. To solve the problem

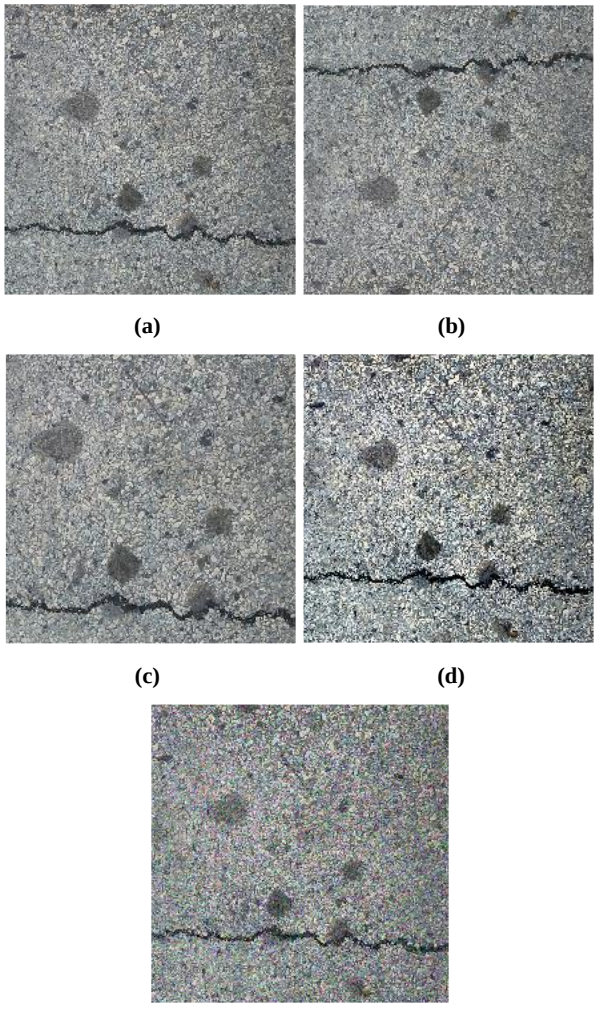
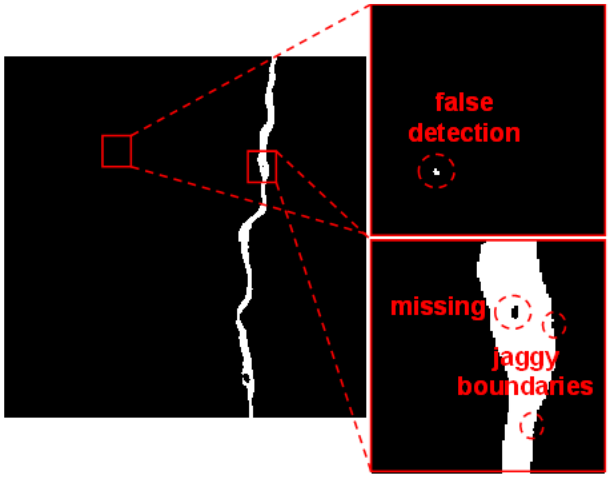


FIGURE 5. Data augmentation by introducing randomness in photographing conditions. (a) Original image. (b) Upside-down flip. (c) Zoomed. (d) Increasing Brightness. (e) Adding gaussian noise.

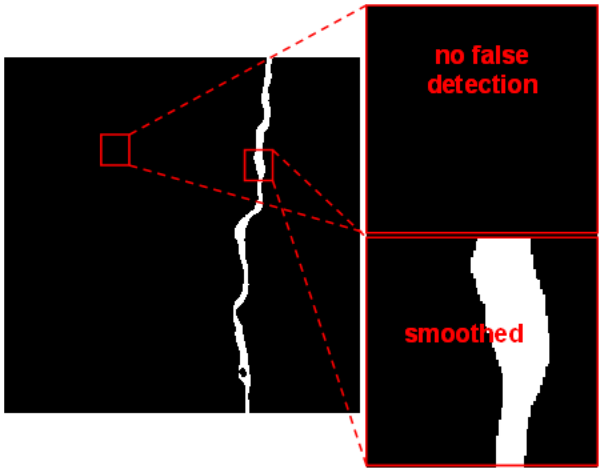
of unbalanced classification, Tversky loss was adopted (Tensorflow [2022]). Defining true positive (the number of positive pixels classified as positive) as tp, false positive (negative pixels classified as positive, false detections) as fp, false negative (positive pixels classified as negative, missing cracks) as fn, Tversky loss Tl was defined below using an unbalance parameter α.

$$Tl = 1 - tp/(tp + \alpha \cdot fp + (1 - \alpha) \cdot fn) \quad (2)$$

$\alpha = 0.3$ is assumed. Tl ranges from 0 to 1. Smaller value is better. The concept of Tversky loss is considering successfully detected crack pixels putting less emphasis on false detections and more emphasis on missing cracks. Because the number of negative pixels was much larger than positive pixels, false detections may be larger than missing cracks after training. In an actual implementation, small values were added to the numerator and denominator to avoid 'divide by zero' problem. When an appropriate loss function is applied,



(a)



(b)

FIGURE 6. Effect of morphology transform. (a) Before morphology transform. Noises in background areas, hole and jaggy boundaries in cracks are observed. (b) After morphology transform. Noises and hole are removed and boundaries are smoothed.

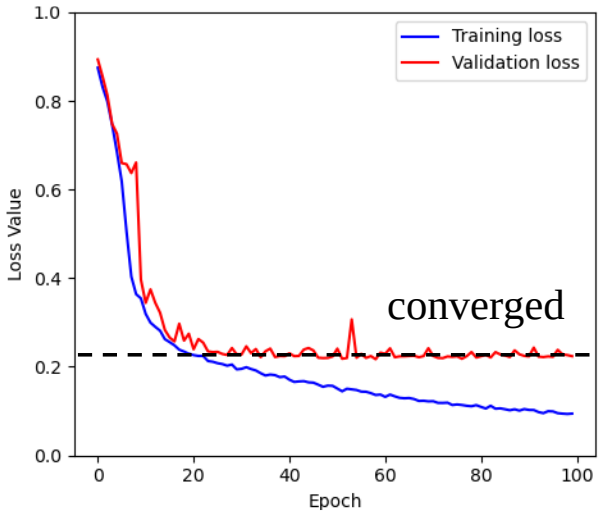


FIGURE 7. Training and validation curves. Loss is Tversky loss considering the balance of falsely detected and missing cracks. Training loss monotonically decreased while validation loss converged.

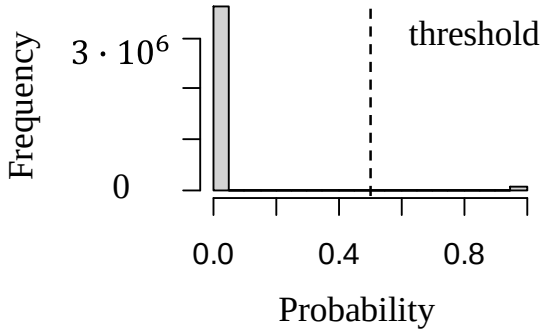


FIGURE 8. Histogram of crack probability for each pixel of Crack #1 image of Crack500. Most pixels are negative and lower than 0.05 while some pixels are positive and higher than 0.995, indicating cracks.

segmentation accuracy is expected to converge to the same value. Stochastic gradient descent (SGD) was used as an optimization method. A learning rate was 0.1 and batch size was 4. Training was manually stopped checking convergence. Training curves were shown in section 4.4 to demonstrate the validity of the training.

4.2 Data augmentation

Data augmentation was studied in the research. Data augmentation increases the number of training images by random transformation. Because deep learning requires large amounts of images, data augmentation is indispensable in some cases. Another effect is data augmentation can introduce variations caused by different measurement conditions such as image shift, rotation, zoom, and brightness changes. Data augmentation helps construct robust deep learning algorithms. Figure 5 shows the augmented images. Figure 5 (a) is an original image. Figure 5 (b) is the flipped image. In the research, left-right flip and top-bottom flip were randomly applied. Figure 5 (c) is the zoomed image of Figure 5 (a). 75 % zoom was randomly applied. Figure 5 (d) is the brighter image after brightness adjustment. Image I' after brightness adjustment β is defined below using RGB values from 0 to 255 of original image I.

$$I' = I \times \beta \tag{3}$$

When $I' > 255$, I' is replaced by 255. Too large β will saturate images. In the research, $\beta = 1.2$ was adopted. Figure 5 (e) is after adding a gaussian noise. The gaussian noise with zero mean and 0.01 variation was randomly applied. All the possible photographing variations were introduced to the training data. The number of training data was doubled after data augmentation.

4.3 Morphology transform

Morphology transform is one of the state-of-the-art image processing techniques to remove noises, smooth boundaries

and extract target features (Haralick *et al.* [1987], Peters [1995]). Morphology transform is based on several spatial filters called structuring elements and applied to binary images. In the research, morphology transform is applied to segmentation results after thresholding probability of cracks. The concept is close to pooling layers of DL. n by n structuring elements M defines the range of neighboring pixels. We adopted $n = 4$.

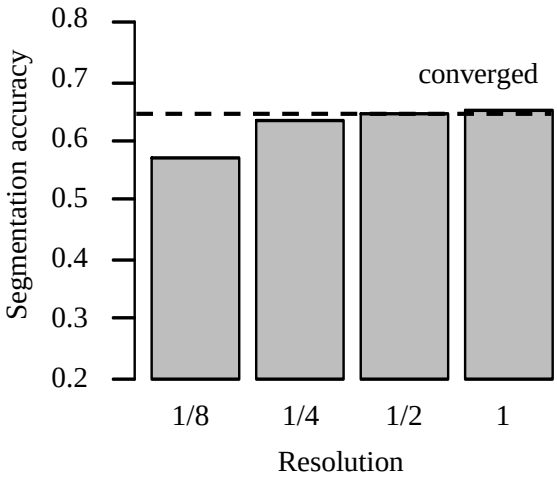
$$M = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \tag{4}$$

'Closing' and 'Opening' operations were sequentially applied in this research. 'Dilation' is to replace a target pixel by 1 when at least one of the neighboring pixels is 1. This is systematically defined by convoluting M to images and judge the output $m > 1$. 'Erosion' is the opposite, to replace the pixel by 0 when at least one of the neighboring pixels is 0, convoluting M and judge $m < n^2$. 'Closing' is 'Erosion' after 'Dilation.' 'Opening' is 'Dilation' after 'Erosion.' 'Closing' is to fill in small holes and 'Opening' is to remove noises. Both operations smooth boundaries of objects.

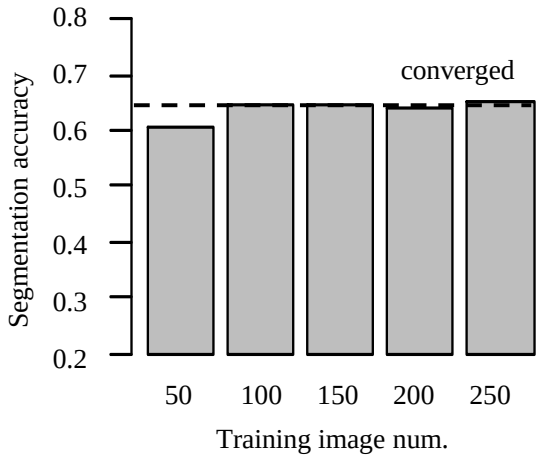
Figure 6 explains the effect of morphology transform using the crack #1 of crack500. As shown in Figure 6 (a), in some cases after segmentation, small noises appeared in background areas and missing holes in cracks. Boundaries of cracks were jaggy. Considering the features of cracks, small speckles should be removed leaving smoothed line-like objects. Figure 6 (b) shows the map after morphology transform. Noises and hole were removed and boundaries were smoothed.

4.4 Segmentation results

Before discussing the effects of U-Net architectures, data augmentation and morphology transform, several important points are shown for a fair comparison. Figure 7 exhibits the example of training and validation curves showing the training process. Loss means Tversky loss defined in section 4.1. All the trainings were successfully accomplished. Training curves decreased monotonically while validation curves converged after 10 minutes to 30 minutes training (30 epochs to 40 epochs). Segmentation accuracy was compared after checking the convergence. The developed architecture outputs a map of the probability of a crack for each pixel. Figure 8 shows the histogram of probabilities of crack #1, one of the constructed crack data by the authors. As shown by the histogram, it is apparent that most background pixels were between 0 and 0.05 and crack pixels were 0.95 and 1. To output binary images, crack maps should be thresholded. From the histogram, the threshold value was not important. In the research, simply the threshold $t = 0.5$ was assumed to distinguish crack and background pixels. In the contexts of image processing and deep learning research, the resolution of images and number of training data were important parameters for high quality crack segmentation. Figure 9 (a) shows the effect of the resolution. Training and test images, and corresponding segmentation



(a)



(b)

FIGURE 9. Effect of characteristics of datasets on segmentation accuracy. (a) Resolution of images. Original image is 1. (b) Number of training images.

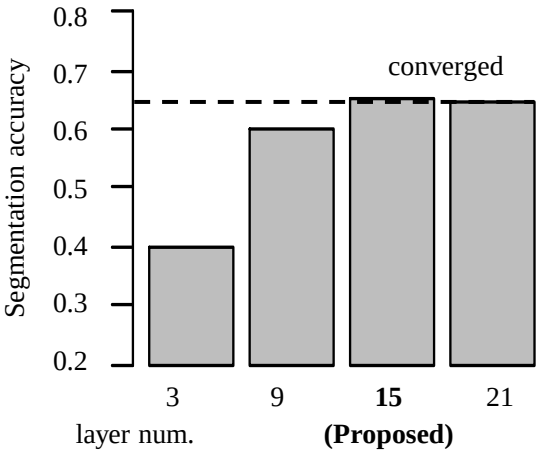


FIGURE 10. Effect of number of layers on segmentation accuracy. It converged at 15 layers.

teacher data were downsampled to produce low resolution data. Segmentation accuracy was converged with certain resolutions. From the result, the resolution was enough with given measurement conditions in terms of crack detection. The resolution depends on the performance of a camera and distance between the road surface and camera. It is suggested that with ordinary smartphones, even a 5 m distance is possible to detect road cracks. However, there may be a problem in matching Lidar data and camera images. The thickness of road cracks is relatively large. To target small hair cracks, finer resolutions may be needed. There is a redundancy in the camera measurement system in terms of road crack detection. Figure 9 (b) shows the effect of the number of the training images. Segmentation accuracy converged with the certain number of images. In deep learning research the number of training data often becomes a bottleneck to conduct a fair comparison of the performance of deep learning architectures. In this research, segmentation accuracy purely depends on the characteristics of the proposed architectures.

The detection of cracks from images by U-Net does not require too deep architectures. Figure 10 explains the effect of the number of U-Net layers on segmentation accuracy. A 15-layer model is the proposed model shown in Figure 4. From Figure 10, segmentation accuracy converged at 15 layers. To extract abstract features requires a certain number of layers. However, the features of cracks, which consisted of bifurcated and fluctuating darker lines are not so various and complicated as general segmentation tasks such as detecting vehicles from ordinary photo images. From Figure 10, the conclusion is that 15 layers were enough.

The proposed algorithm, combination of a U-Net architecture, data augmentation, and morphology transform is the most accurate compared with previous deep learning architectures. Table 2 shows the comparison of segmentation accuracy and calculation time. The claim of the previous research is that top-down and bottom-up feature pyramids consider multiple scale contexts to improve segmentation accuracy. For a comparison, Fcn-8s is a typical fully convolutional network (FCN) composed of VGG16 convolution architectures. VGG16 is one of the up-to-date deep learning architectures, which can use parameters learned for general image classification tasks (Krizhevsky *et al.* [2012], Simonyan *et al.* [2014], Szegedy *et al.* [2015]). The previous research improved segmentation accuracy. The proposed U-Net shows higher accuracy by naturally introducing multiple scale features using skip connections. Calculation time is halved by an efficient feature map integration scheme and the same level as FCN. Furthermore, segmentation accuracy improved by data augmentation (proposed 1). It can be stated that this is not the effect of increasing training data as discussed in Figure 9 but introducing random measurement condition changes in training data to construct a robust deep learning algorithm. Calculation time is the same because an application phase is the same. U-Net, data augmentation and morphology transform shows the highest segmentation accuracy (proposed 2). 0.004 difference is not negligible because considering 200 test images with 1500 pixels by 2500 pixels, 750 million pixels, segmentation accuracy improved by 3 million pixels. This is owing to the reduction of small noises, holes and smoothed crack shapes. Morphology transform took 0.3 s per image to apply a sequence of spatial filters. Calculation time is not important in the case of on-site and in-house analysis. However, to apply the algorithm to real-time videoframe analysis, U-Net with data augmentation may be the first choice. We adopted morphology transform hereafter.

TABLE 2. Comparison of segmentation accuracy and calculation time of proposed and previous algorithms.

Model	Segmentation accuracy (-)	Calculation time per image (s)
Fcn-8s (Yang *et al.* (2020))	0.577	0.101 (NVIDIA GeForce GTX TITAN X)
FPHBN (Yang *et al.* (2020), previous)	0.635	0.197 (the same as above)
U-Net	0.649	0.114 (NVIDIA GeForce GTX 1080ti)
U-Net+Aug. (proposed 1)	**0.677**	**0.114 (the same as above)**
U-Net+Aug. +Morph. (proposed 2)	**0.681**	**0.445 (the same as above)**

Figure 11 shows the segmentation results of Crack500 by the proposed algorithm (proposed 2). It is difficult to completely reproduce the learned parameters of the previous research. Simple U-Net without data augmentation and morphology transform outperformed the previous algorithm. Therefore, Figure 11 compares the proposed algorithm with simple U-Net. Figure 11 (a) and Figure (b) exhibit the crack #1 and #2. OIS of majority of the images were from 0.7 to 0.9. The predicted segmentation images by the proposed algorithm matched well with the true segmentation images while U-Net caused noises and missing parts of the cracks. Data augmentation and morphology transform improved segmentation accuracy. The results do not depend on the color patterns of pavements, and shapes and thicknesses of cracks. Figure 11 (c) picks up the case in which OIS is lower than 0.7 and not showing large differences between the U-Net and proposed algorithm. Figure 11 (c) shows the portion of an alligator crack with relatively complicated bifurcations. More complicated the geometries of cracks are, lower OIS is because of pixel-level shifts. However, from Figure 11 (c), the prediction results clearly show the reasonable geometry of the crack.

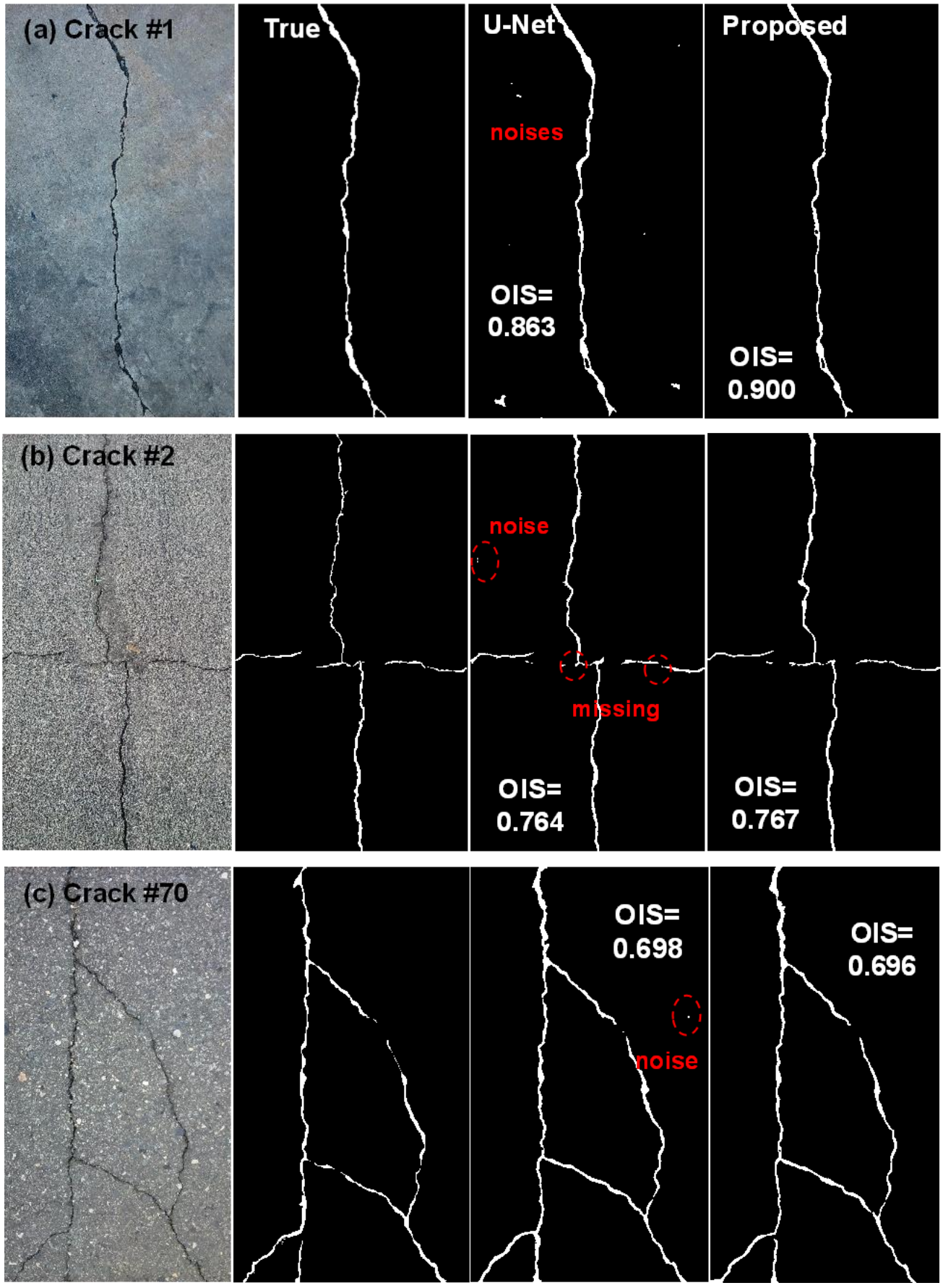
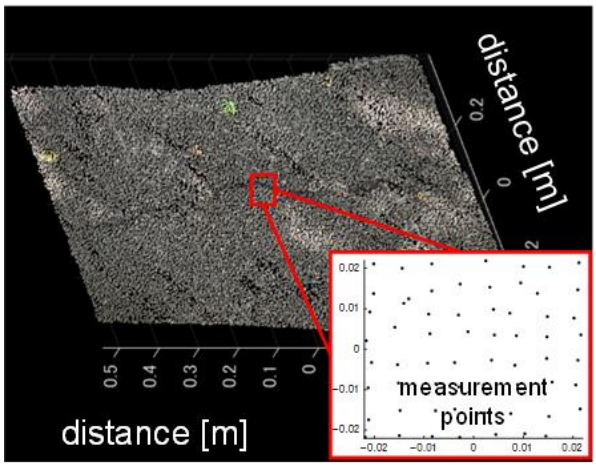


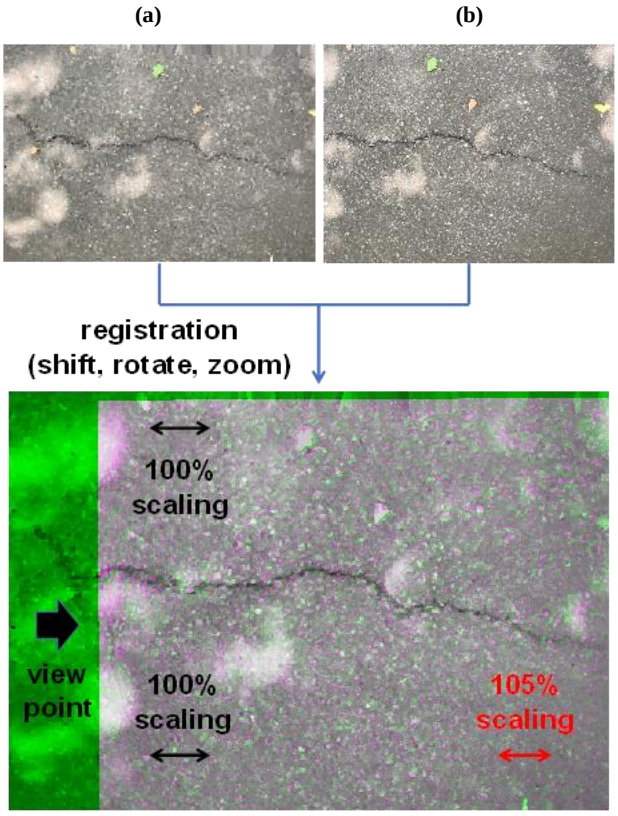
FIGURE 11. Road crack segmentation results Crack500. Left column is images. Second column is ground truth. Third column is U-Net for comparison. Right column is proposed algorithm. (a) Crack #1. (b) Crack #2. (c) Crack #70. Various crack shapes and asphalt pavement color patterns are observed.



(a)

(b)

FIGURE 12. Lidar data. (a) 3D point cloud data of crack #1 collected by the authors. (b) Resampled Lidar color data. Resolution of image is not high. However, characteristic features such as cracks, leaves, light conditions, and pavement color patterns are observable.



(c)

FIGURE 13. Registration process. (a) Lidar color data of Figure 12 (b). (b) Camera image. (c) Registered images. Scaling and viewpoint are also shown. Registration is successfully accomplished by MATLAB R2021a.

5 HOW TO QUANTIFY SEGMENTATION RESULTS BY LIDAR DATA

5.1 Matching camera images to Lidar data

This section explains how to match images to Lidar data to quantitatively evaluate cracks, called registration. The unique point is that the data is multimodal. Generally, registration requires not a small calculation cost, about one to two minutes per image depending on the sizes of images. When quantitative estimation is not needed or the scales of images are known using fixed systems, the registration process can be skipped. Because measurement conditions cannot be constant using smartphones, registration should be conducted.

Before registration, Lidar data should be resampled to make a constant resolution. Figure 12 (a) shows the Lidar data of crack #1 of the constructed data by the authors. Road surfaces are assumed to be a plane, i.e. no large altitude changes. We set the minimum 1 cm resolution laser scanning. However, as shown in the right bottom picture of Figure 12 (a), typical Lidar measurement points are far from uniform. In the research, grid data was produced simply by referring to the nearest point at each grid position. Figure 12 (b) shows the resampled Lidar color data with 1 mm resolutions in horizontal directions. 1 mm is considering the variations of the positions of the measurement points. The one-pixel distance of Lidar color data exactly corresponds to 1 mm. Lidar color data is useful from Figure 12 (b) though the resolution is too low to evaluate cracks. Because App refers to camera images to construct Lidar color data, the black line of the crack can be observed.

Figure 13 explains the registration process. Figure 13 (a) shows the Lidar data and Figure 13 (b) shows the camera image. At this moment, we do not know what the one-pixel distance of the camera image is. The camera image was matched to the lidar color data allowing shift, rotate and zoom operations. Shift in two directions dx and dy, rotation and zooming center positions z_x, z_y, rotation angle θ and zoom magnification factor MF are the parameters, called similarity transform. If the problem only considers shift (translation) or shift and rotation (rigid), the number of the parameters can be reduced. Considering shear transform is called affine transform. Because the registration process is unstable and needs a large calculation cost depending on the number of

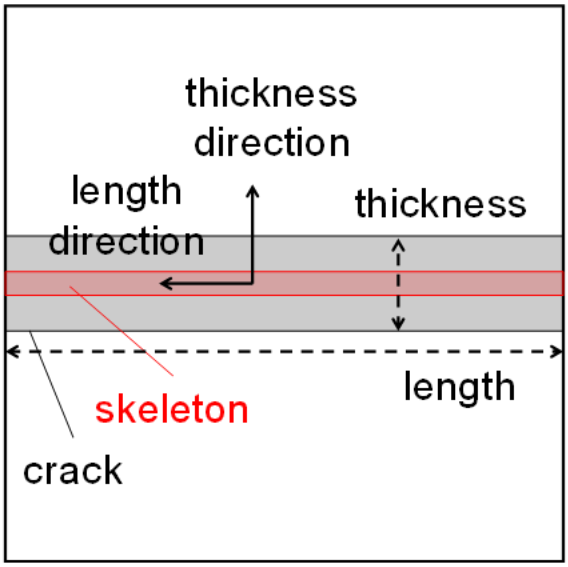


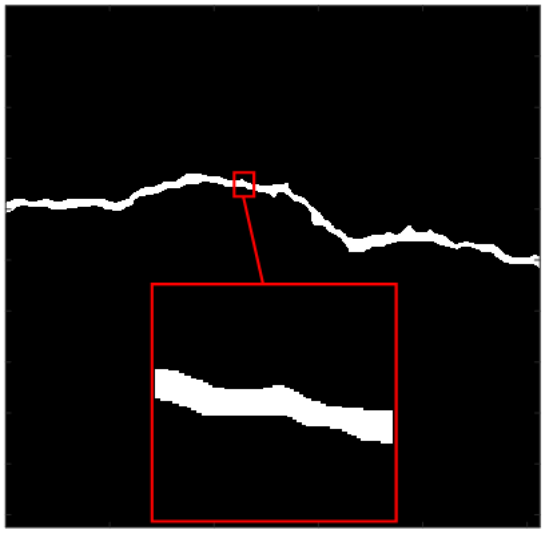
FIGURE 14. Diagram of terms introduced in this research. Length and thickness directions of cracks are estimated to quantitatively evaluate cracks.

unknown parameters, considering possible measurement condition variations, similarity transform was adopted.

Registration was introduced by MATLAB R2021a image processing toolbox (MATLAB, [2021]). Color images were converted to gray scale images calculating the mean value of RGB channels and normalized to consider multimodal data. 6 parameters were optimized by taking the correlation value of the two images. The local maximum was searched by assigning random initial parameters to narrow search regions. The parameters are: $InitialRadius = 0.001$; $Epsilon = 1.5 \cdot 10^{-6}$; $GrowthFactor = 1.05$; $MaxIteration = 300$. Figure 13 (c) is the result. Registrations were successfully conducted without divergences and unreasonable estimation results.

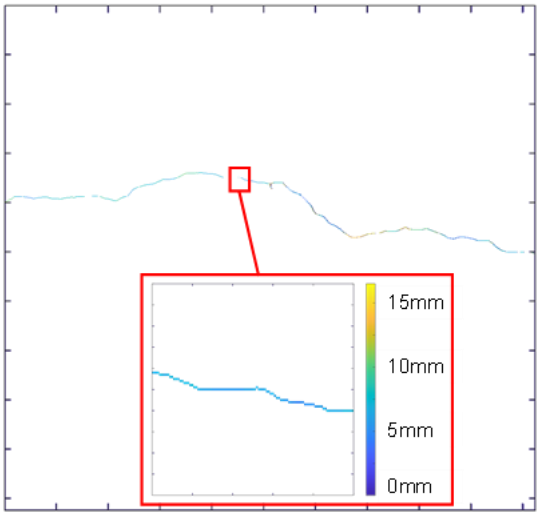
Two important points should be discussed here. The first point is, in the case of Figure 13, shadows and fallen leaves may help register images. However, the most important features are cracks and pavement color patterns as shown in section 5.3. At least the method can be applied to asphalt pavement images. The second point is from Figure 13 (c), zooming is essential to consider both the scales of images and inhomogeneous resolutions. Shift and rotation will not change the scales of cracks. Zooming will estimate the scale of the image. Moreover, depending on a view point, regions near the point will not change and distant will be enlarged. For example, considering magnification in a horizontal direction, as shown in Figure 13 (c), assuming the left side of the image is a 100 % scale, the top, 0.5 m distant is the same scale and right side, 1 m distant is 105 % scales. This 5 % additional enlargement directly corresponds to the 5 % error of the geometry estimation. This scaling is reasonable because the smartphone was placed on the left side of the crack. An image should be enlarged around a camera position.

After registration, camera images can be upsampled to make 1 mm uniform resolutions. In the research, to reduce the



(a)

(b)



(c)

FIGURE 15. Estimating crack thicknesses. (a) Segmentation result. (b) Skeleton. (c) Estimation result. A map was output to show thickness distributions.

calculation cost, camera images were first input to the segmentation algorithm and segmentation results were upsampled. The two methods show the same results.

5.2 Estimating lengths and thicknesses

Provided segmentation results with known pixel distances, the lengths and thicknesses of cracks are estimated. Figure 14 denotes the terms introduced in this section. Figure 15 exhibits the process of estimating the thickness.

Thicknesses were estimated at each position of a crack. Segmentation results are composed of fluctuating and bifurcating lines as shown in Figure 15 (a). Skeletonizing operation is applied to segmented cracks to obtain skeletons. Skeletonizing is similar to the 'thinning' and 'erosion' operations of morphology transform. The pixels of the boundaries of an object were eroded as many as possible holding the number and topology of the object. As shown in Figure 15 (b), a skeleton was a thin one-pixel line passing the center of the crack. At each pixel on skeletons, 4 direction continuities were calculated; in vertical, horizontal, left-top to right-bottom, right-top to left bottom directions, the numbers of neighboring pixels, continuities were counted. The direction which offers the largest continuity is a principal direction, in other words, length direction of the crack. A perpendicular direction is defined as a thickness direction. At each point of the skeleton, the continuity in a thickness direction was calculated and converted to a millimeter unit referring to the pixel distances. In diagonal directions, $\sqrt{2}$ should be multiplied. As shown in Figure 15 (c), thicknesses were successfully estimated at each position to show the map of a thickness distribution.

Lengths were estimated by counting the number of the pixels of skeletons. $\sqrt{2}$ should be multiplied to diagonal cracks also here using the information of the principal direction at each pixel. This diagonal factor is not negligible to estimate both length and thickness. Skeleton pixels were merged to understand the number of cracks in an image.

The final results were the segmentation maps of cracks, number of cracks, length of each crack and thicknesses along the crack in colormaps. This information is precise to enable road administrators to evaluate road surface conditions and plan repair works. This method is also useful for screening target sections to prioritize repair plans.

5.3 Geometry estimation results

The proposed method successfully segmented and quantitatively evaluate road cracks of the constructed validation data by the authors. Figure 16, Figure 17 and Table 3 summarized the estimation results. The left column of Figure 16 is the images of the four cracks. The center column is the sketches with the measured total crack lengths and thicknesses at the three points on each crack. In this research we picked up the total 12 points to measure the thicknesses to validate the estimation algorithm. The right column is the segmentation results with the estimated total length and thicknesses. The sketches were considered as ground truths. Table 3 lists the true and estimated total lengths of the four cracks. Figure 17 shows the accuracy of thickness estimation by a scatter plot, true versus estimated thicknesses.

Crack #1 of Figure 16 (a) shows a typical crack pattern with a 580 mm length and 5 mm thicknesses. The image also includes a fallen leaf and sunlight filtered by trees as obstacles. Crack #2 shows a portion of an alligator crack with a 990 mm total length and relatively larger thicknesses from 10 mm to 20 mm. The crack was filled with vegetations and sands. The difficulty of crack #2 is a rough noisy pavement color pattern. Crack #3 is three fragments of cracks with a 655mm total length and 7mm thicknesses. Crack #4 is a forked crack with a 795mm total length and 5mm thicknesses. The pavement color patterns of crack #4 were different between the left and light sides of the image. The four cracks have different geometries, thicknesses, and pavement color patterns to demonstrate the effectiveness of the proposed method.

The segmentation results clearly represent the geometries of the cracks in all the four cases. One fluctuating line of crack #1, crossing lines of crack # 2 and crack #4, and three fragments of crack #3 are apparent from predictions. The segmentation results are so precise that they may be further analyzed to judge the types and severities of cracks by using image processing methods such as pattern analysis and calculating fractal dimensions. The results were also apparent for non-practitioners.

The quantitative parameters of all the four cracks were accurately estimated with a centimeter-order accuracy in terms of lengths and millimeter-order accuracy in terms of thicknesses. From Table 3, in the best case the ground truth length was 990mm while the estimated length was 970mm, only 2 % error. The errors of the estimated lengths were within 10 % in all the cases. The total length of the four crack cases was 3020 mm, while the estimated length was 2876 mm, only 5 % error. It is impossible to achieve a 5 mm accuracy in measuring the lengths of actual cracks. All the results underestimated the crack lengths. The possible reasons are that the developed segmentation algorithm tended to underestimate the areas of the cracks and because of the bias of the inspector. The estimated lengths were useful in a practical sense.

From Figure 16 right column, the thicknesses were reflected in the segmentation maps and accurately estimated by the proposed algorithm. Because of the number of the data, an adjusted coefficient of determination was estimated. From Figure 17, the adjusted coefficient of determination was high, about 0.98. The thicknesses varied from 3 mm to 20 mm and estimated within ± 1 mm errors. It is difficult to accurately measure the thicknesses of cracks with submillimeter accuracy. The resolutions of the original images were around 0.5 mm considering the number of the pixels and actual distances. The resolutions of the Lidar data were 1 cm. Matching results may be intermediate, around 1mm. Considering this point, estimating the thicknesses of cracks on concrete surfaces may be difficult because typical thicknesses of concrete cracks are submillimeter. In terms of road crack detection, from Figure 16, the bold lines of crack #2 were apparent, indicating the severity of the crack.

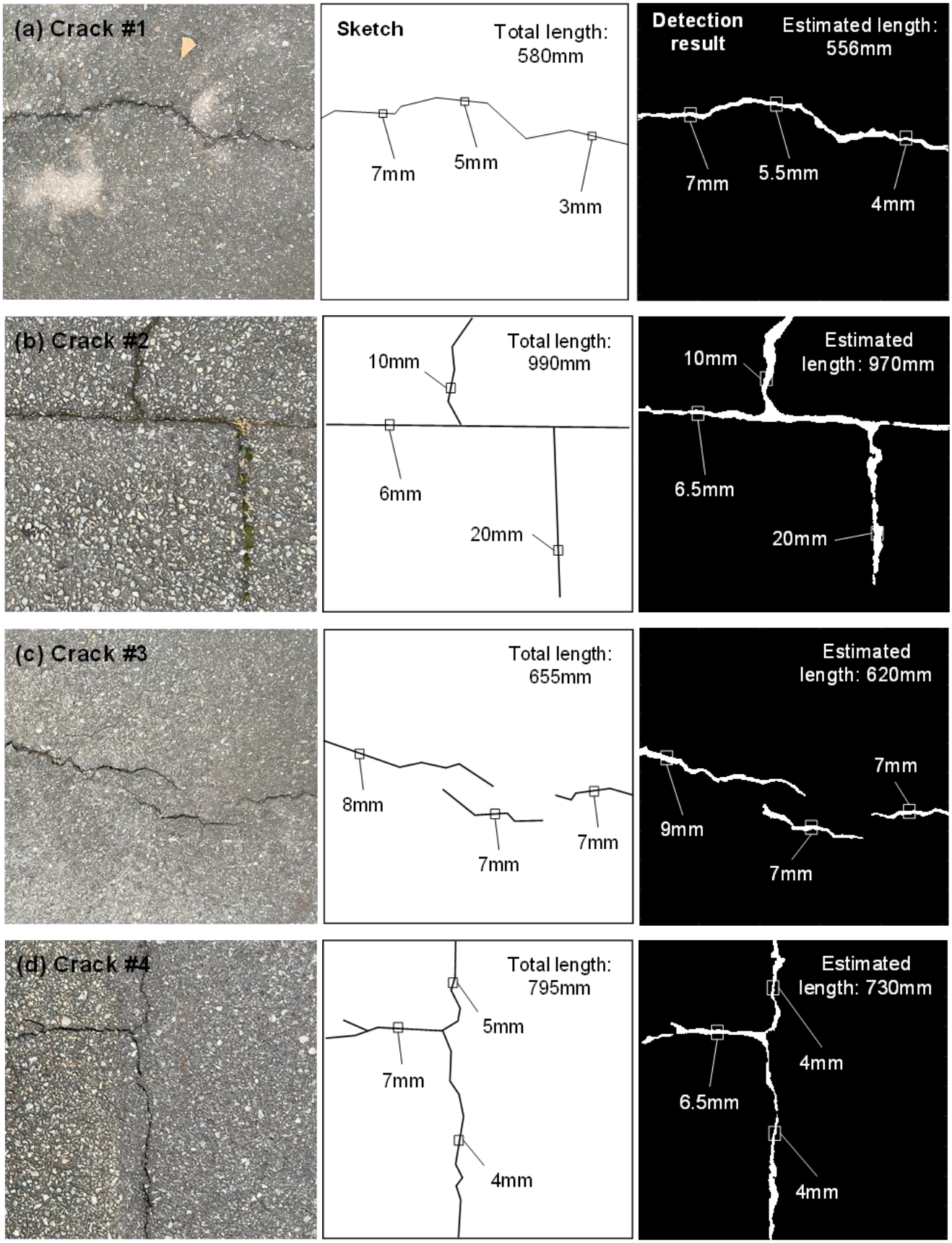


FIGURE 16. Road crack quantification results of four cases collected by the authors. Left column is crack images. Center column is sketches considered as ground truths. Right column is evaluation results utilizing Lidar data. (a) Crack #1 is a typical crack including a leaf with different lighting conditions. (b) Crack #2 is an alligator crack with larger thicknesses. Noises of pavement color patterns are large. (c) Crack #3 is three fragments of cracks. (d) Crack #4 is an alligator crack. Left and right sides, two pavement color patterns exist. Various crack shapes and pavement color patterns are observed.

TABLE 3. Comparison of true and estimated lengths. Errors were within 10 % in all the cases.

Crack No.	True length	Estimated (Error)
No. 1	580 mm	556 mm (-4.1%)
No. 2	990 mm	970 mm (-2.0%)
No. 3	655 mm	620 mm (-5.3%)
No. 4	795 mm	730 mm (-8.2%)
Total	**3020 mm**	**2876 mm (-4.8%)**

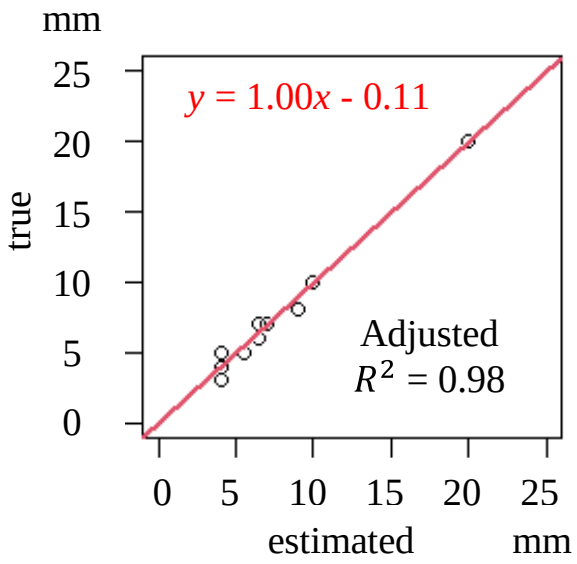


FIGURE 17. Scatter plot of true versus estimated thicknesses. Three thicknesses for each crack, total twelve thicknesses were plotted as black circles. Adjusted coefficient of determination R^2 was 0.98 with least square approximation as red line. R^2 shows the accuracy of estimated thicknesses.

6 DISCUSSIONS

From the discussions above, the conclusion is that the developed segmentation algorithm showed the highest segmentation accuracy. The proposed quantitative evaluation algorithm estimated crack lengths with centimeter-order accuracy and thicknesses with millimeter-order accuracy, providing impactful results. The following discussions are to show the limitations for the future applications of the method and directions for future works.

In terms of the segmentation algorithm, the advantage of our algorithm compared with the previous machine learning algorithms is a sequence of optimized spatial filters and integrated features automatically learned from a large database to offer the highest accuracy. The uniqueness of the algorithm compared with other deep learning algorithms for road crack detection is data augmentation and morphology transform. Data augmentation naturally introduced the variations of photographing conditions. However, the algorithm tends to falsely detect the boundaries of manhole covers, shadows, and patchworks of repaired pavements. It may be effective to introduce road appendages and shadows to training data to reduce false detections. A cautionary point is increasing the images of the target objects will decrease the ratio of crack data, which may cause failures in training processes. Gaussian mixture may naturally introduce target objects to existing images (Maeda *et al.* [2021]). Boundaries of repair works are black lines and similar to cracks. They may be distinguished by line detection algorithms such as Hough transform after segmentation. Crack databases may be increased to adjust to various pavement color patterns in different countries.

Morphology transform removed noises and holes, and smoothed boundaries. The algorithm assumes smoothed line-like features of cracks. On the other hand, U-Net focusses on the local features of cracks. Considering the continuities of cracks in the whole image, i.e. lengths, thicknesses and fractal dimensions may reduce false detections.

It is indispensable to utilize Lidar data to conduct accurate quantification. However, a registration step needed some time. The Lidar system acquires Lidar point cloud data, camera images and geometrical relationships between the two data. However, a typical Lidar measurement App outputs Lidar data with a color attribution. Directly extracting relationship information may omit the registration process. It is also better to improve the resolution of Lidar data for further accurate evaluations. Our algorithm assumes an ordinary Lidar App to realize a simple implementation.

The appealing point of our method is the evaluation of lengths and thicknesses of cracks is possible only by one shot of an ordinary smartphone. Measurement took one to two seconds. Segmentation took less than one second. Registration took not a small time but can be finished within one to two minutes. The method is applicable to cracks on road and wall surfaces. Lidar data can reconstruct detailed geometries of surfaces. Therefore, the method is amenable to complicated surfaces, i.e. the edges of structures. The applicability of the algorithm on concrete cracks is not clear. Buildings, steel bridge elements, manufactured products and concrete structures may be future applications. The distance can be over 5 m. However, an inclined view is not favorable to detect cracks in distant regions and conduct the accurate registration. This method is also applicable to pothole and rutting detections and evaluations, which is left as future works. By collecting damage cases from users, our deep learning algorithm can further be improved.

The aim of the research is the integration of multimodal data, Lidar data and camera images to realize complete and accurate road damage detections. Our research group is developing GPR and seismic wave systems for subsurface

sensing. The temporal accumulation of multimodal data in large scale areas will realize statistical analysis in time and space dimensions. A future sensing platform may be proposed offering up-to-date accurate condition assessment results, which strongly helps road administrators' decisions.

7 Conclusions

In this research, an automatic, accurate, quantitative and simple evaluation method of road cracks is proposed using smartphone images and Lidar data. The algorithm consists of two steps. The trained segmentation algorithm based on U-Net, data augmentation and morphology transform showed the highest segmentation accuracy on the common crack database. After registering camera images to Lidar data by similarity transform after resampling, the algorithm estimated crack lengths and thicknesses, defining crack directions using skeletons. 95% length accuracy and 0.98 coefficient of determination for thickness estimation were achieved for the data with various crack shapes and asphalt pavement color patterns collected by the authors. The algorithm showed the persuasive segmentation and quantification results even for non-practitioners.

The possible future works are that we are developing road pothole and rutting detection algorithms based on smartphone images and Lidar data. Cracks on complicated surfaces such as buildings, steel bridge elements, manufactured products and concrete structures may be future applications. False detections may be reduced by introducing road appendage training data and considering the geometrical features of cracks such as lengths, thicknesses, line patterns and fractal dimensions. It is expected that the registration process can be facilitated, and databases can be increased by developing own Lidar measurement platform.

REFERENCES

American Society of Civil Engineers (ASCE) (2021). Infrastructure Report Card. https://infrastructurereportcard.org/cat-item/roads-infrastructure. Accessed on: 2022.9.23.

Apple inc. (2020). iPhone 12 Pro technical specifications. https://support.apple.com/kb/SP831?viewlocale=en_US&locale=en_US. Accessed on: 2022.9.23.

Bang, S., Park, S., Kim, H., & Kim, H. Encoder-decoder network for pixel-level road crack detection in black-box images. Computer-Aided Civil and Infrastructure Engineering. 34(8). 713-727.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York. USA. Springer. 225-284.

Chun, P., & Hashimoto, K. (2015). Development of Semi-Automatic Asphalt Pavement Crack Detection System using Image Processing and Machine Learning Approach. Japan Society of Civil Engineers Ser. E1. 71(3). I_31-I_38.

Chun, P., Izumi, S., & Yamane, T. (2021). Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. Computer Aided Civil and Infrastructure Engineering. 36(1). 61-72.

Cotič, P., Kolarič, D., Bosiljkov, V. B., Bosiljkov, V., & Jagličić, Z. (2015), Determination of the applicability and limits of void and delamination detection in concrete structures using infrared thermography. NDT&E International. 74. 87-93.

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction. 99. 52-58.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Bosco, C. D., Walsh, S., Saltukoglu, D., Tay, T. L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., & Ronneberger, O. (2019). U-Net: deep learning for cell counting, detection, and morphometry. Nature Methods. 16. 67-70.

Fujita, Y., & Hamamoto, Y. (2011). A robust automatic crack detection method from noisy concrete surfaces. Machine Vision and Applications. 22. 245-254.

Girshick, R. (2015). Fast R-CNN. IEEE International Conference on Computer Vision. Santiago. Chile. Dec.

Goodfellow, I., Bengio, Y, & Courville, A. (2016). *Deep learning*. Massachusetts. USA. the MIT Press. 326-365.

Guan, H., Li, J., Yu, Y., Chapman, M., & Wang, C. (2015). Automated Road Information Extraction from Mobile Laser Scanning Data. IEEE Transactions on Intelligent Transportation Systems. 16(1). 194-205.

Harada, T. (2017). *Image Recognition* (in Japanese). Tokyo. Japan: Kodansha. 156-193.

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image Analysis Using Mathematical Morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-9(4). 532-550.

Hoang, N. D., & Nguyen, Q. L. (2019). A novel method for asphalt pavement crack classification based on image processing and machine learning. Engineering with Computers. 35. 487-498.

Huang, Z., Chen, W., Abir Al-Tabbaa, A., & Brilakis, I. (2022). NHA12D: A New Pavement Crack Dataset and a Comparison Study of Crack Detection Algorithms. IEEE Conference on Computer Vision and Pattern Recognition. New Orleans. USA. Jun.

Huyan, J., Li, W., Tighe, S., Xu, Z., & Zhai, J. (2020). CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection. Structural Control and Health Monitoring. 27(8). e2551.

Jarvis, R. A. (1983). A Perspective on Range Finding Techniques for Computer Vision. IEEE Transactions on pattern analysis and machine intelligence. PAMI-5(2). 122-139.

Ji, A., Xue, X., Wang, Y., Luo, X., & Xue, W. (2020). An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. Automation in Construction. 114.103176.

Kim, B., & Cho, S. (2019). Image-based concrete crack assessment using mask and region-based convolutional neural network. Structural Control and Health Monitoring. 26. e2381.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 25. 1097-1105.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision. Amsterdam. Netherland. Oct.

Liu, Y., & Gao, M. (2022). Detecting cracks in concrete structures with the baseline model of the visual characteristics of images. Computer-Aided Civil and Infrastructure Engineering. Early Access.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. Computer-Aided Civil and Infrastructure Engineering. 33(12). 1127-1141.

Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., & Omata, H. (2021). Generative adversarial network for road damage detection. Computer-Aided Civil and Infrastructure Engineering. 36(1). 47-60.

Maguire, M., Dorafshan, S., & Thomas, R. J. (2018). SDNET2018. https://digitalcommons.usu.edu/all_datasets/48. Accessed on: 2022.9.23.

MATLAB. (2022). imregtform: Estimate geometric transformation that aligns two 2-D or 3-D images (Image Processing Toolbox). https://jp.mathworks.com/help/images/ref/imregtform.html?lang=en. Accessed on: 2022.9.23.

Ministry of Land, Infrastructure, Transport and Tourism (MLIT). (2017). Road pavement inspection manual (in Japanese). www.mlit.go.jp/road/sisaku/yobohozen/tenken/yobo3_1_10.pdf. Accessed on: 2022.9.23.

Ministry of Land, Infrastructure, Transport and Tourism (MLIT). (2021). Annual road report and statistics (in Japanese). https://www.mlit.go.jp/road/ir/ir-data/tokei-nen/2020/nenpo02.html. Accessed on: 2022.9.23.

Mizutani, T., Yamaguchi, T., Kudo, T., Yamamoto, K., Ishida, T., Nagata, Y., Kawamura, H., Tokuno, T., Suzuki, K., & Yamaguchi, Y. (2022). Quantitative evaluation of peeling and delamination on infrastructure surfaces by laser signal and image processing of 3D point cloud data. Automation in Construction. 133.104023.

Nakamura, H., Kameyama, S., Sakuraba, S., & Yokota, Y. (2020). Examination on crack evaluation index of pavement. Journal of the Japan Society of Civil Engineers Ser. E1. 74(1). 58-69.

Nishikawa, T., Yoshida, J., Sugiyama, T., & Fujino, Y. (2012). Concrete Crack Detection by Multiple Sequential Image Filtering. Computer-Aided Civil and Infrastructure Engineering. 27(1). 29-47.

NVIDIA. (2022). GEFORCE GTX 1080 ti. www.nvidia.com/en-gb/geforce/products/10series/geforce-gtx-1080-ti/#. Accessed on: 2022.9.23.

PassMark Software. (2022). High End Video Cart Chart. https://www.videocardbenchmark.net/high_end_gpus.html. Accessed on: 2022.9.23.

Peters, R. A. (1995). A New Algorithm for Image Noise Reduction Using Mathematical Morphology. IEEE Transactions on Image Processing. 4(5). 554-568.

Popovic, A., Fuente, M., Engelhardt, M., & Radermacher, K. (2007). Statistical validation metric for accuracy assessment in medical image segmentation. International Journal of Computer Assisted Radiology and Surgery. 2. 169-181.

Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., & Parvardeh, H. (2016). Automated Crack Detection on Concrete Bridges. IEEE Transactions on Automation Science and Engineering. 13(2). 591-599.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition. Nevada. USA. Jun.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39(6). 1137-1149.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. IEEE Conference on Computer Vision and Pattern Recognition. Columbus. USA. Jun.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic Road Crack Detection Using Random Structured Forests. IEEE Transactions on Intelligent Transportation Systems. 17(12). 3434-3445.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition. Boston. USA. Jun.

TensorFlow. (2022). TensorFlow. www.tensorflow.org. Accessed on: 2022.9.23.

Vidal, T., Castel, A., & François, R. (2004). Analyzing crack width to predict corrosion in reinforced concrete. Cement and Concrete Research. 34(1). 165-174.

Yamaguchi, T., Mizutani, T., Tarumi, M., & Su, D. (2019). Sensitive damage detection of reinforced concrete bridge slab by “time-variant deconvolution” of SHF-band radar signal. IEEE Transactions on Geoscience and Remote Sensing. 57(3). 1478-1488.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2020). Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. IEEE Transactions on Intelligent Transportation Systems. 21(4). 1525-1535.

Zalama, E., Bermejo, J. G. G., Medina, R., & Llamas, J. (2014) Road Crack Detection using Visual Features Extracted by Gabor Filters. Computer-Aided Civil and Infrastructure Engineering. 29(5). 342-358.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City. USA. Jun.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. Pattern Recognition Letters. 33(3). 227-238.

3D Scanner App. (2022). https://3dscannerapp.com. Accessed on: 2022.9.23