Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images: Road damage detection and classification

Article *in* Computer-Aided Civil and Infrastructure Engineering · June 2018

DOI: 10.1111/mice.12387

5 authors, including:

Hiroya Maeda
The University of Tokyo
21 PUBLICATIONS **831** CITATIONS
SEE PROFILE

Yoshihide Sekimoto
The University of Tokyo
178 PUBLICATIONS **2,365** CITATIONS
SEE PROFILE

Toshikazu Seto
Komazawa University
49 PUBLICATIONS **828** CITATIONS
SEE PROFILE

Takehiro Kashiyama
The University of Tokyo
45 PUBLICATIONS **906** CITATIONS
SEE PROFILE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **0** (2018) 1–15

INDUSTRIAL APPLICATION

Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images

Hiroya Maeda, Yoshihide Sekimoto*, Toshikazu Seto, Takehiro Kashiyama & Hiroshi Omata

Institute of Industrial Science, The University of Tokyo, Tokyo, Japan

Abstract: *Research on damage detection of road surfaces using image processing techniques has been actively conducted. This study makes three contributions to address road damage detection issues. First, to the best of our knowledge, for the first time, a large-scale road damage data set is prepared, comprising 9,053 road damage images captured using a smartphone installed on a car, with 15,435 instances of road surface damage included in these road images. Next, we used state-of-the-art object detection methods using convolutional neural networks to train the damage detection model with our data set, and compared the accuracy and runtime speed on both, using a GPU server and a smartphone. Finally, we demonstrate that the type of damage can be classified into eight types with high accuracy by applying the proposed object detection method. The road damage data set, our experimental results, and the developed smartphone application used in this study are publicly available (https://github.com/sekilab/RoadDamageDetector/).*

1 INTRODUCTION

During the period of high economic growth in Japan from 1954 to 1973, infrastructure such as roads, bridges, and tunnels were constructed extensively; however, because many of these were constructed more than 50 years ago (Ministry of Land, Infrastructure, Transport and Tourism, 2016), they are now aged, and the number of structures that are to be inspected is expected to increase rapidly in the next few decades. In addition, the discovery of the aged and affected parts of infrastructure has thus far depended solely on the expertise of veteran field engineers. However, owing to the increasing demand for inspections, a shortage of field technicians (experts) and financial resources has resulted in many areas. In particular, the number of municipalities that have neglected conducting appropriate inspections owing to the lack of resources or experts has been increasing (Tomiyama et al., 2013). The United States also has similar infrastructure aging problems (AASHTO, 2008). Indeed, the prevailing problems in infrastructure maintenance and management are likely to be experienced by countries all over the world. Considering this negative trend in infrastructure maintenance and management, it is evident that efficient and sophisticated infrastructure maintenance methods are urgently required.

In response to this problem, many methods to efficiently inspect infrastructure, especially road conditions, have been studied, such as methods using laser technology or image processing. Moreover, there are several studies using neural networks for civil engineering problems in the 11 years from 1989 to 2000 (Adeli, 2001). Furthermore, recently, computer vision and machine learning techniques have been successfully applied to automate road surface inspection (Chun et al., 2015; Zalama et al., 2014; Jo and Ryu, 2015).

However, thus far, with respect to methods of inspections using image processing, we believe these methods suffer from three major disadvantages:

(1) There is no common data set for a comparison of results; in each research, the proposed method is evaluated using its own data set of road damage images. Motivated by the field of general object recognition, wherein large common data sets such as ImageNet (Deng et al., 2009) and PASCAL VOC (Everingham et al., 2015) exist, we believe there is a need for a common data set on road scratches.

(2) Although current state-of-the-art object detection methods use end-to-end deep learning

*To whom correspondence should be addressed. E-mail: *sekimoto@iis.u-tokyo.ac.jp*.

DOI: 10.1111/mice.12387

techniques, no such method exists for road damage detection.

(3) Though road surface damage is distinguished into several categories (in Japan, eight categories according to the Road Maintenance and Repair Guidebook 2013; JRA, 2013), many studies have been limited to the detection or classification of damage in only the longitudinal and lateral directions (Chun et al., 2015; Zalama et al., 2014; Zhang et al., 2016; Akarsu et al., 2016; Maeda et al., 2016).

Therefore, it is difficult for road managers to apply these research results directly in practical scenarios. Considering these disadvantages, in this study, we develop a new, large-scale road damage data set, and then train and evaluate a damage detection model that is based on the state-of-the-art convolutional neural network (CNN) method.

The contributions of this study are as follows.

(1) We created and released 9,053 road damage images containing 15,435 instances of damage. The data set contains the bounding box of each class for the eight types of road damage. Each image is extracted from an image set created by capturing pictures of a large number of roads obtained using a vehicle-mounted smartphone. The 9,053 images of the data set contain a wide variety of weather and illuminance conditions. In addition, in assessing the type of damage, the expertise of a professional road administrator was employed, rendering the data set considerably reliable.

(2) Using our developed data set, we evaluated the state-of-the art object detection method based on deep learning and generated benchmark results. All the trained models are also publicly available on our Web site (https://github.com/sekilab/RoadDamageDetector/).

(3) Furthermore, we showed that the type of damage from among the eight types can be identified with high accuracy.

The rest of the article is organized as follows. In Section 2, we discuss the related works. Details of our new data set are presented in Section 3. The experimental settings are explained in Section 4. Then, the results and the discussion are provided in Sections 5 and 6, respectively. Finally, Section 7 concludes the article.

2 RELATED WORKS

2.1 Road damage detection using image processing

Several attempts have been made to develop a method for analyzing road properties by using a combination of recordings by in-vehicle cameras and image processing technology to more efficiently inspect a road surface. For example, a previous study proposed an automated asphalt pavement crack detection method using image processing techniques and a naive Bayes-based machine-learning approach (Chun et al., 2015). In addition, a pothole detection system using a commercial black-box camera has been previously proposed (Jo and Ryu, 2015). In recent times, it has become possible to quite accurately analyze the damage to road surfaces using deep neural networks (Zhang et al., 2016; Maeda et al., 2016; Zhang et al., 2017; Fan et al., 2018). For instance, Zhang et al. (2017) introduced CrackNet, which predicts class scores for all pixels. However, such road damage detection methods focus only on the determination of the existence of damage. Though some studies do classify the damage based on types—for example, Zalama et al. (2014) classified damage types vertically and horizontally, and Akarsu et al. (2016) categorized damage into three types, namely, vertical, horizontal, and crocodile—most studies primarily focus on classifying damages between a few types. There are other studies that detect blurry road markings (Kawano et al., 2017), and classify the cracks and sealed cracks (Zhang et al., 2018). Therefore, for a practical damage detection model for use by municipalities, it is necessary to clearly distinguish and detect different types of road damage; this is because, depending on the type of damage, the road administrator needs to follow different approaches to rectify the damage.

Furthermore, the application of deep learning for road surface damage identification has been proposed by a few studies, for example, studies by Maeda et al. (2016) and Zhang et al. (2016). However, the method proposed by Maeda et al. (2016), which uses 256 × 256 pixel images, identifies the damaged road surfaces, but does not classify them into different types. In addition, the method of Zhang et al. (2016) identifies whether damage occurred exclusively using a 99 × 99 patch obtained from a 3,264 × 2,448 pixel image. Further, a 256 × 256 pixel damage classifier is applied using a sliding window approach (Felzenszwalb et al., 2010) for 5,888 × 3,584 pixel images to detect cracks on the concrete surface (Cha et al., 2017). In these studies, classification methods are applied to input images and damage is detected. Recently, it has been reported that object detection using end-to-end deep learning is more accurate and has a faster processing speed than using a combination of classification methods; this will be discussed in detail in Section 2.4. As an example of a method using end-to-end deep learning performing better than tradition methods, white line detection based on end-to-end deep learning using OverFeat (Sermanet et al., 2013) outperformed a previously proposed empirical method (Huval et al., 2015). However, to the best of our knowledge, no example of the application of end-to-end deep learning method for road damage detection exists. It is important to note that classification refers to labeling an image rather than an object, whereas detection means assigning an image a label and identifying the object's coordinates as exemplified by the ImageNet competition (Deng et al., 2009). The term "end-to-end" indicates that input and output relationships are trained directly with a single model.

Therefore, considering this, we apply the end-to-end object detection method based on deep learning to the road surface damage detection problem, and verify its detection accuracy and processing speed. In particular, we examine whether we can detect eight classes of road damage by applying state-of-the-art object detection methods (discussed in Section 2.4) with the newly created road damage data set (explained in Section 3). Although many excellent methods have been proposed, such as segmentation of cracks on concrete surfaces (O'Byrne et al., 2014; Nishikawa et al., 2012) and metallic surfaces (Chen et al., 2017), our research uses an object detection method. Indeed, although there is research that uses deep learning to evaluate the stability of structures using sensor data (Rafiei and Adeli, 2017, 2018; Lin et al., 2017; Rafiei et al., 2017), in this article, we concentrate on detecting road surface damage using image processing.

2.2 Road damage detection using smartphones

In general, vehicles designed specifically for road inspection are expensive. Meanwhile, mobile devices such as smartphones have made remarkable progress in recent years, and examples of road inspection using smartphone sensors are increasingly common. Using a smartphone is advantageous insofar as it is possible to inspect the road surface cheaply and exhaustively. For example, Buttlar and Islam (2014) proposed a method to measure the flatness of a road using the accelerometer of a smartphone installed in a car. Furthermore, Casas-Avellaneda and López-Parra (2016) proposed a method that visualizes (on a map) potholes detected by smartphone sensors. In addition, Mertz et al. (2014) proposed a method to handle road images acquired by on-board smartphones installed on cars that operate on a daily basis, such as general passenger automobiles, buses, and garbage trucks, to detect road surface damage with an external laptop.

To the best of our knowledge, however, there is no research on processing road images acquired by smartphones to detect road damage. Therefore, we demonstrate that using end-to-end deep learning is feasible for processing such images.

2.3 Image data set of road surface damage

Although an image data set of the road surface exists, called the KITTI data set (Geiger et al., 2013), it is primarily used for applications related to automated driving. There is also the GAP data set for road damage detection with features of around 2,000 high-resolution images with manually annotated damage (six classes) (Eisenbach et al., 2017). To the best of our knowledge, the GAP data set is the only publicly available data set for road damage detection. In all the studies focusing on road damage detection described in Section 2.1, the researchers independently proposed unique methods using acquired road images. Therefore, a comparison between the methods presented in these studies is difficult.

Furthermore, according to Mohan and Poobal (2017), there are few studies that construct damage detection models using real data, and 20 of these studies use road images taken directly from above the road. For instance, the images of the GAP data set were taken from above the road. In fact, it is difficult to reproduce the road images taken from directly above, because doing so involves installing a camera outside the car body, which, in many countries, is a violation of the law; in addition, it is costly to maintain a dedicated car solely for road images.

Therefore, we have developed a data set of road damage images using the road images captured using a smartphone on the dashboard of a general passenger car; in addition, we made this data set publicly available. There are five times more road images in this data set than in the GAP data set. Moreover, we show that road surface damage can be detected with considerably high accuracy even with images acquired by employing such a simple method.

2.4 Object detection system

In a CNN-based object detection method such as R-CNN (Region-based Convolutional Neural Networks; Girshick et al., 2014) and Fast R-CNN (Girshick, 2015), it is necessary to obtain the object candidate region in advance by using another method, such as selective search (Uijlings et al., 2013) or BING (Binarized Normed Gradients; Cheng et al., 2014). For this reason, the process is slow and the judgment accuracy is relatively low, insofar as it is a two-stage process (i.e., the candidate area is detected and the detected area is classified by a CNN). On the other hand, Faster R-CNN (Ren et al., 2015) makes it possible to train the model end-to-end, and the accuracy of determination and the execution speed can be improved by using the Region Proposal Network, which performs object candidate region detection. Furthermore, rather than cropping features from the same layer where the region proposals are predicted—as in the case of the Faster R-CNN method—the R-FCN (Region-based Fully Convolutional Networks) method proposed by Dai et al. (2016) crops from the last layer of features prior to prediction. This approach of pushing cropping to the last layer minimizes the amount of per-region computation that must be performed. Dai et al. (2016) showed that their R-FCN model (using Resnet 101) could achieve accuracy comparable to Faster R-CNN, and often at faster running speeds. Although the processing speed has been greatly improved by the above method, the computational load is somewhat large when processing images from modern mobile devices.

YOLO (You Only Look Once) (Redmon et al., 2016; Redmon and Farhadi, 2017) is an object detection framework that can achieve high mean average precision (mAP) and speed. In addition, YOLO can predict the region and class of objects with a single CNN. An advantageous feature of YOLO is that its processing speed is considerably fast, because it solves the problem as a mere regression, detecting objects by considering background information. The YOLO algorithm outputs the coordinates of the bounding box of the object candidate and the confidence of the inference after receiving an image as input. Furthermore, SSD (Single Shot MultiBox Detector; Liu et al., 2016) is an object detection framework that uses a single feed-forward convolutional network to predict classes directly and anchor offsets without requiring a second stage per proposal classification operation. The key feature of this framework is the use of multiscale convolutional bounding box outputs attached to multiple feature maps at the top of the network. With this key feature, SSD is fast and has fewer errors than YOLO. In this research, SSD is adopted as a training algorithm for processing images from a mobile terminal.

2.5 Feature extractor

In all these object detection systems, a convolutional feature extractor as a base network is applied to the input image to obtain high-level features. The selection of the feature extractor is considerably important because the number of parameters and layers, the type of layers, and other properties directly affect the performance of the detector. Darknet-19 (Redmon and Farhadi, 2017) is a base model of the YOLO framework. The model has 19 convolutional layers and five maxpooling layers. Furthermore, VGG 16 (Simonyan and Zisserman, 2014) is a CNN with a total of 16 layers consisting of 13 convolution layers and three fully connected layers proposed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. This model achieved good results in ILSVRC and COCO 2015 (classification, detection, and segmentation) considering the depth of the layers.

Resnet (He et al., 2016), which refers to Deep Residual Learning, is a structure for deep learning, particularly for CNNs, that enables high-precision learning in a very deep network; it was released by Microsoft Research in 2015. Accuracy beyond human ability was obtained by learning images with 154 layers. Resnet achieved an error rate of 3.57% with the ImageNet test set and won the first place in the ILSVRC 2015 classification task. Although Resnet has extremely high accuracy, it takes considerable time to process, because the layer is deep. On the other hand, Inception V2 (Ioffe and Szegedy, 2015) and Inception V3 (Szegedy et al., 2016) can increase the depth and breadth of the network without increasing the number of parameters or the computational complexity, by introducing so-called inception units. By adding an inception layer, the calculation process is reduced. MobileNet (Howard et al., 2017) has succeeded in further suppressing the amount of calculation and has been shown to achieve accuracy comparable to VGG-16 on ImageNet with only 1/30th of the computational cost and model size. MobileNet is designed for efficient inference in various mobile vision applications. Its building blocks are depthwise-separable convolutions that factorize a standard convolution into a depthwise convolution and a 1 × 1 convolution, effectively reducing both the computational cost and the number of parameters. These five feature extractors are widely used in the field of computer vision. According to Huang et al. (2017), Inception V2 and MobileNet offer the fastest processing speeds with relatively high determination accuracy. Therefore, we selected Inception V2 and MobileNet to evaluate our results (see Section 5).

3 PROPOSED DATA SET

In this section, we describe our proposed new data set, including how the data were obtained, how it was annotated, its contents, and issues related to privacy.

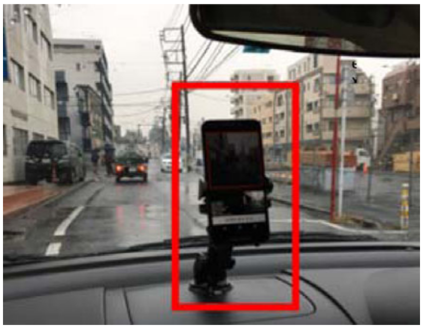


Fig. 1. Installation setup of the smartphone on the car. It is mounted on the dashboard of a general passenger car. Our developed application can capture a photograph of the road surface approximately 10 m ahead, which indicates that this application can photograph images while traveling on the road without leakage or duplication when the car moves at an average speed of about 40 km/h (about 10 m/s) if photographing every second. In addition, it can detect road damages in 1.5 seconds with high accuracy (see Section 5).

3.1 Data collection

Thus far, in the study of damage detection on the road surface, images are either captured from above the road surface or using on-board cameras on vehicles. When models are trained with images captured from above, the situations that can be applied in practice are limited, considering the difficulty of capturing such images.

In contrast, when a model is constructed with images captured from an on-board vehicle camera, it is easy to apply these images to train the model for practical situations. For example, using a readily available camera like on smartphones and general passenger cars, any individual can easily detect road damages by running the model on the smartphone or by transferring the images to an external server and processing it on the server.

We selected seven local governments in Japan (Ichihara city, Chiba city, Sumida ward, Nagakute city, Adachi ward, Muroran city, and Numazu city) and cooperated with the road administrators of each local government to collect 163,664 road images. We traveled through every municipality covering approximately 1,500 km in total. Seven municipalities have snowy areas and urban areas that are very diverse in terms of regional characteristics such as the weather and fiscal constraints.

We installed a smartphone (LG Nexus 5X) on the dashboard of a car, as shown in Figure 1, and photographed images of 600 × 600 pixels once per second. The reason we select a photographing interval of 1 second is because it is possible to photograph images while traveling on the road without leakage or duplication when the average speed of the car is approximately 40 km/h (or approximately 10 m/s). For this purpose, we created a smartphone application that can capture images of the roads and record the location information once per second; this application is also publicly available on our Web site.

3.2 Data category

Table 1 lists the different damage types and their definition. In this article, each damage type is represented with a class name such as D00. Each type of damage is illustrated in the examples in Figure 2.

As can be seen from the table, the damage types are divided into eight categories. First, the damage is classified into cracks or other corruptions. Then, the cracks are divided into linear cracks and alligator cracks (crocodile cracks). Other corruptions include not only potholes and rutting, but also other road damage such as blurring of white lines.

To the best of our knowledge, no previous research covers such a wide variety of road damages, especially in the case of image processing. For example, the method proposed by Jo and Ryu (2015) detects only potholes in

Table 1
Road damage types in our data set and their definitions

Damage type			*Detail*	*Class name*
Crack	Linear crack	Longitudinal	Wheel mark part	D00
			Construction joint part	D01
		Lateral	Equal interval	D10
			Construction joint part	D11
	Alligator crack		Partial pavement, overall pavement	D20
Other corruption			Rutting, bump, pothole, separation	D40
			Crosswalk blur	D43
			White line blur	D44

Source: Road Maintenance and Repair Guidebook 2013 (JRA, 2013) in Japan.
Note: In reality, rutting, bumps, potholes, and separations are different types of road damage, but it is difficult to distinguish these four types using images. Therefore, they were classified as one class, namely, D40.

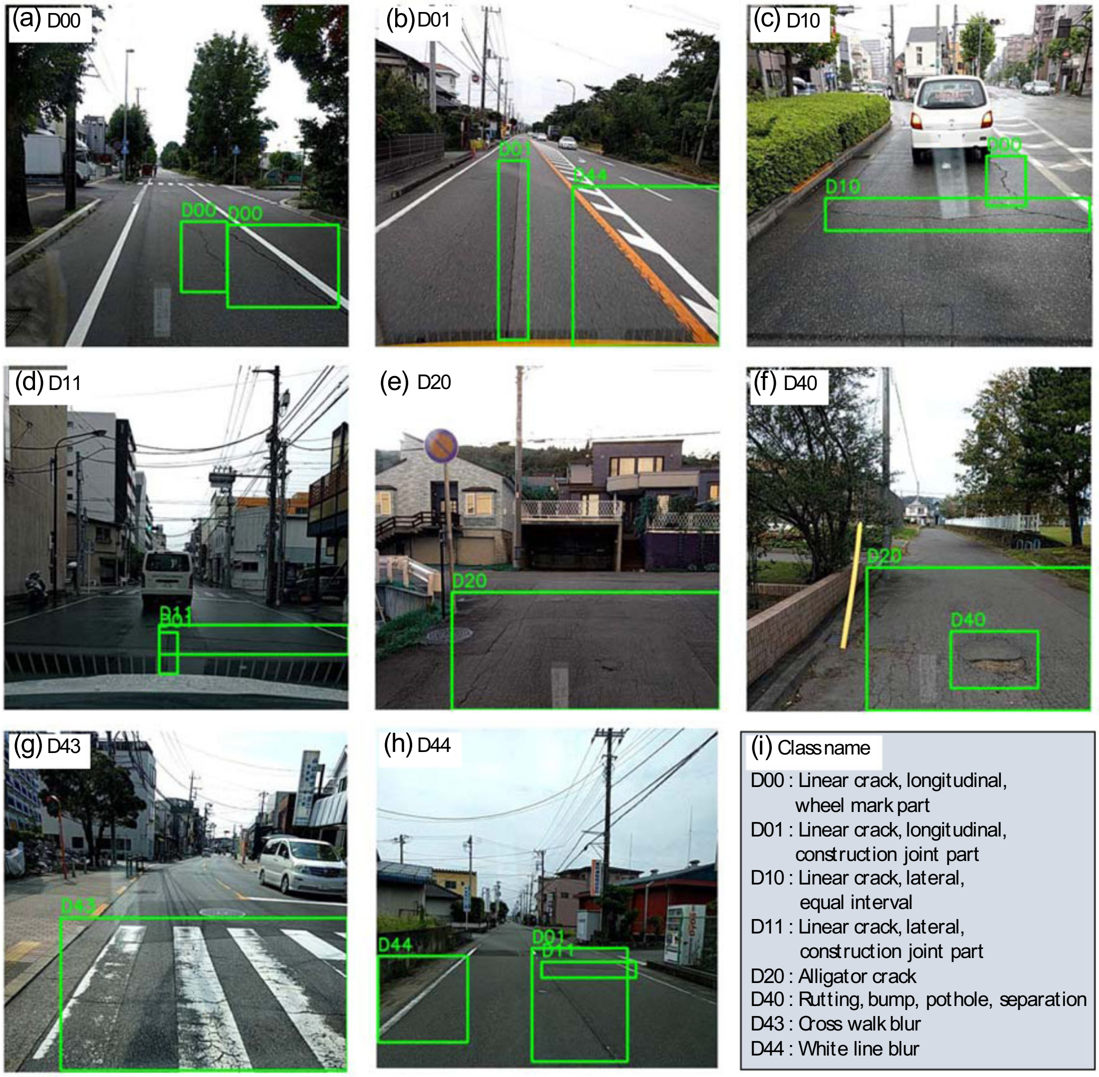

(a) D00
D00
D00
(b) D01
D01
D44
(c) D10
D00
D10
(d) D11
D11
D01
(e) D20
D20
(f) D40
D20
D40
(g) D43
D43
(h) D44
D44
D01
D11
(i) Class name
D00 : Linear crack, longitudinal, wheel mark part
D01 : Linear crack, longitudinal, construction joint part
D10 : Linear crack, lateral, equal interval
D11 : Linear crack, lateral, construction joint part
D20 : Alligator crack
D40 : Rutting, bump, pothole, separation
D43 : Cross walk blur
D44 : White line blur

Fig. 2. Sample images of our data set: (a)–(h) correspond to each one of the eight categories, and (i) shows the legend. Our benchmark contains 163,664 road images and of these, 9,053 images include cracks. These 9,053 images were annotated with class labels and bounding boxes. The images were captured using a smartphone camera in realistic scenarios.

D40, and that of Zalama et al. (2014) classifies damage types exclusively as longitudinal and lateral, whereas the method proposed by Akarsu et al. (2016) categorizes damage types into longitudinal, lateral, and alligator cracks. Further, the previous study using deep learning (Maeda et al., 2016; Zhang et al., 2016) only detects the presence or absence of damage. Further, although the GAPs data set (Eisenbach et al., 2017) treats the crack as a crack, in our data set, the cracks are classified into five types.

3.3 Data annotation

The collected images were then annotated manually. We illustrate our annotation pipeline in Figure 3. Because our data set format is designed in a manner similar to the PASCAL VOC (Everingham et al., 2010, 2015), it is easy to apply it to many existing methods used in the field of image processing.

3.4 Data statistics

Our data set is composed of 9,053 labeled road damage images. Of these 9,053 images, 15,435 bounding boxes of damage are annotated. Figure 4 shows the number of instances per label that were collected in each municipality. We photographed a number of road images in various regions of Japan, but could not avoid biasing some of the data. For example, damages such as

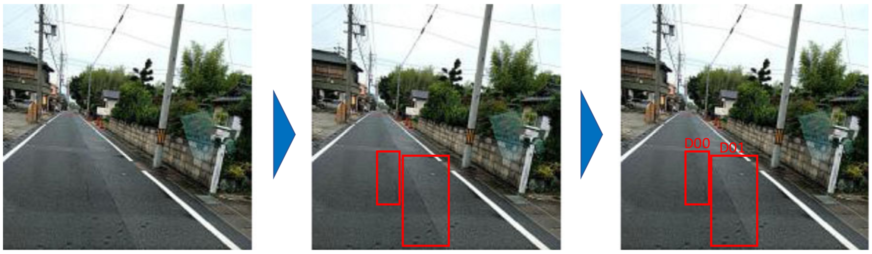
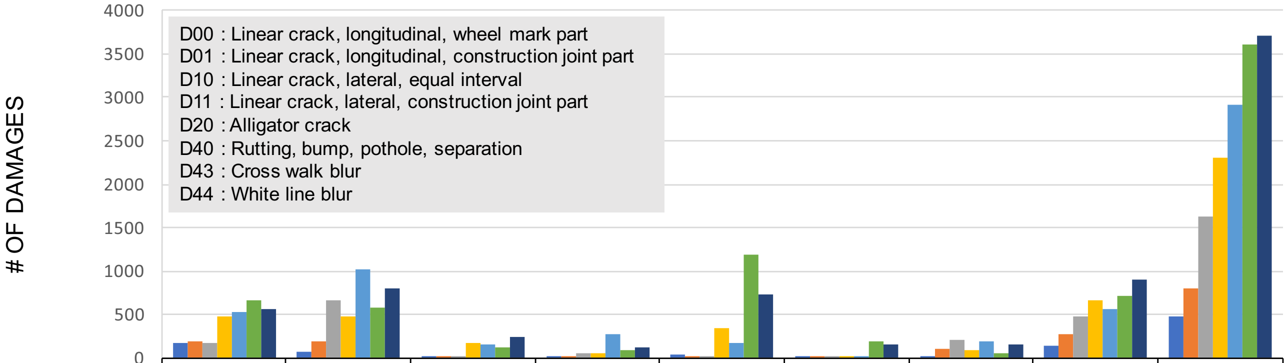


Fig. 3. Annotation pipeline. First, the bounding box is drawn. Then, the class label is attached. Next, expert road managers checked them.



	D00	D01	D10	D11	D20	D40	D43	D44	TOTAL
■ Ichihara city	175	71	18	9	43	8	20	138	482
■ Chiba city	183	187	13	12	27	3	104	267	796
■ Sumida ward	168	660	20	61	21	19	201	482	1632
■ Nagakute city	482	477	169	58	351	14	90	659	2300
■ Adachi ward	529	1013	153	279	172	11	191	567	2915
■ Muroran city	671	574	124	88	1192	189	50	712	3600
■ Numazu city	560	807	245	129	735	165	161	908	3710
TOTAL	2768	3789	742	636	2541	409	817	3733	15435

Fig. 4. Number of damage instances in each class in each municipality. We can see that the distribution of damage type differs for each local government. For example, in Muroran city, there are many D20 damages (1,192 damages) compared to other municipalities. This is because Muroran city is a snowy zone, therefore, alligator cracks tend to occur during the thaw of snow.

D40 pose a more significant danger, and therefore, road managers repair these damages as soon as they occur; thus, there are not many instances of D40 in reality. In many studies, the blurring of white lines is not considered to be damage; however, in this study, white line blur is also considered as damage. In summary, our new data set includes 9,053 damage images and 15,435 damage bounding boxes. The resolution of the images is 600 × 600 pixels. The area and the weather in the area are diverse, and thus, the data set closely resembles the real world. We used this data set to evaluate the damage detection model.

3.5 Privacy matters

Our data set is openly accessible by the public. Therefore, considering issues with privacy, based on visual inspection, when a person's face and a car license plate are clearly reflected in the image, they are blurred out.

4 EXPERIMENTAL SETUP

Based on a previous study in which many neural networks and object detection methods were compared in detail (Huang et al., 2017), among the state-of-the-art object detection methods, the SSD using Inception V2 and SSD using MobileNet are those with relatively small CPU loads and low memory consumption, even while maintaining high accuracy. However, it is important to note that the results of the abovementioned research were obtained using the COCO data set (Lin et al., 2014). Because we believe that an object detection method that can be executed on a smartphone (or a

Table 2
Detection and classification results for each class

	D00	*D01*	*D10*	*D11*	*D20*	*D40*	*D43*	*D44*
Recall of SSD Inception V2	0.22	0.60	0.10	0.05	0.68	0.03	0.81	0.62
Precision of SSD Inception V2	0.73	0.84	0.99	0.95	0.73	0.67	0.77	0.81
Accuracy of SSD Inception V2	0.78	0.80	0.94	0.92	0.85	0.95	0.95	0.83
Recall of SSD MobileNet	0.40	0.89	0.20	0.05	0.68	0.02	0.71	0.85
Precision of SSD MobileNet	0.73	0.64	0.99	0.95	0.68	0.99	0.85	0.66
Accuracy of SSD MobileNet	0.81	0.77	0.92	0.94	0.83	0.95	0.95	0.81

small computational resource) is desirable, in this study, we trained the model using the SSD Inception V2 and SSD MobileNet frameworks.

We analyze the cases of applying the SSD using Inception and SSD using MobileNet to our data set in detail.

4.1 Parameter settings

In the object detection algorithm using deep learning, the parameters learned from the data are enormous; in addition, the number of hyper parameters set by humans is large. The parameter setting in the case of each algorithm is described below.

4.1.1 SSD using Inception V2. We followed the methodology mentioned in the original paper (Liu et al., 2016). The initial learning rate is 0.002, which is reduced by a learning rate decay of 0.95 per 10,000 iterations. The input image size is 300 × 300 pixels, which indicates that the original images are resized from 600 × 600 to 300 × 300.

4.1.2 SSD using MobileNet. As in the previous case, we followed the methodology mentioned in the original paper (Liu et al., 2016) as well. Similar to Huang et al. (2017), we initialize the weights with a truncated normal distribution with a standard deviation of 0.03. The initial learning rate is 0.003 with a learning rate decay of 0.95 every 10,000 iterations. The input image size in this case is 300 × 300 pixels as well.

4.2 Training and evaluation

We conducted training and evaluation using our data set. For our experiment, the data set was randomly divided in a ratio of 8:2; the former part was set as training data, and the latter as evaluation data. Thus, the training data included 7,240 images, and the evaluation data had 1,813 images. During training, the images were randomly flipped horizontally for data augmentation, and this was done with a probability of 0.5.

5 RESULTS

In our experiment, training was performed on a PC running the Ubuntu 16.04 operating system with an NVIDIA GRID K520 GPU and 15 GB RAM memory using TensorFlow. In the evaluation, the Intersection Over Union threshold was set to 0.5. The detected samples are illustrated in Figures 5 and 6.

We compared the results provided by the SSD Inception V2 and SSD MobileNet. These results are listed in Table 2. Although D01 and D44 were detected with relatively high recall and precision, the value of recall is low in the case of D11 and D40; this can be attributed to the number of training data (see Figure 4). On the contrary, D43 was detected with high recall and precision even though the number of training data is small; this is because D43 (blur of the pedestrian crossing) occupies a large proportion in the image and the feature is clear (i.e., stripped pattern). When paying attention to the value of Recall, MobileNet exceeds Inception in six categories, except D40 and D43. Overall, the SSD MobileNet yields better results.

To better understand the detection results, we conducted error analysis. The errors were classified as the false positives and false negatives. Typical examples of false positives are shown in Figure 7. There were examples where side gutters and manholes were judged to be damage, and judgments that the reflection from windshield wipers was blur on a crosswalk. There were also cases where a blurred sign on the road was judged as damage. On the other hand, there were almost no cases where shadows on the road were judged to be damage. We believe that as many shadows on the road were included as negative examples in the training data, it was rare to judge a shadow as damage. On the other hand, as the training data did not contain many blurred signs on the road or windshield wipers,

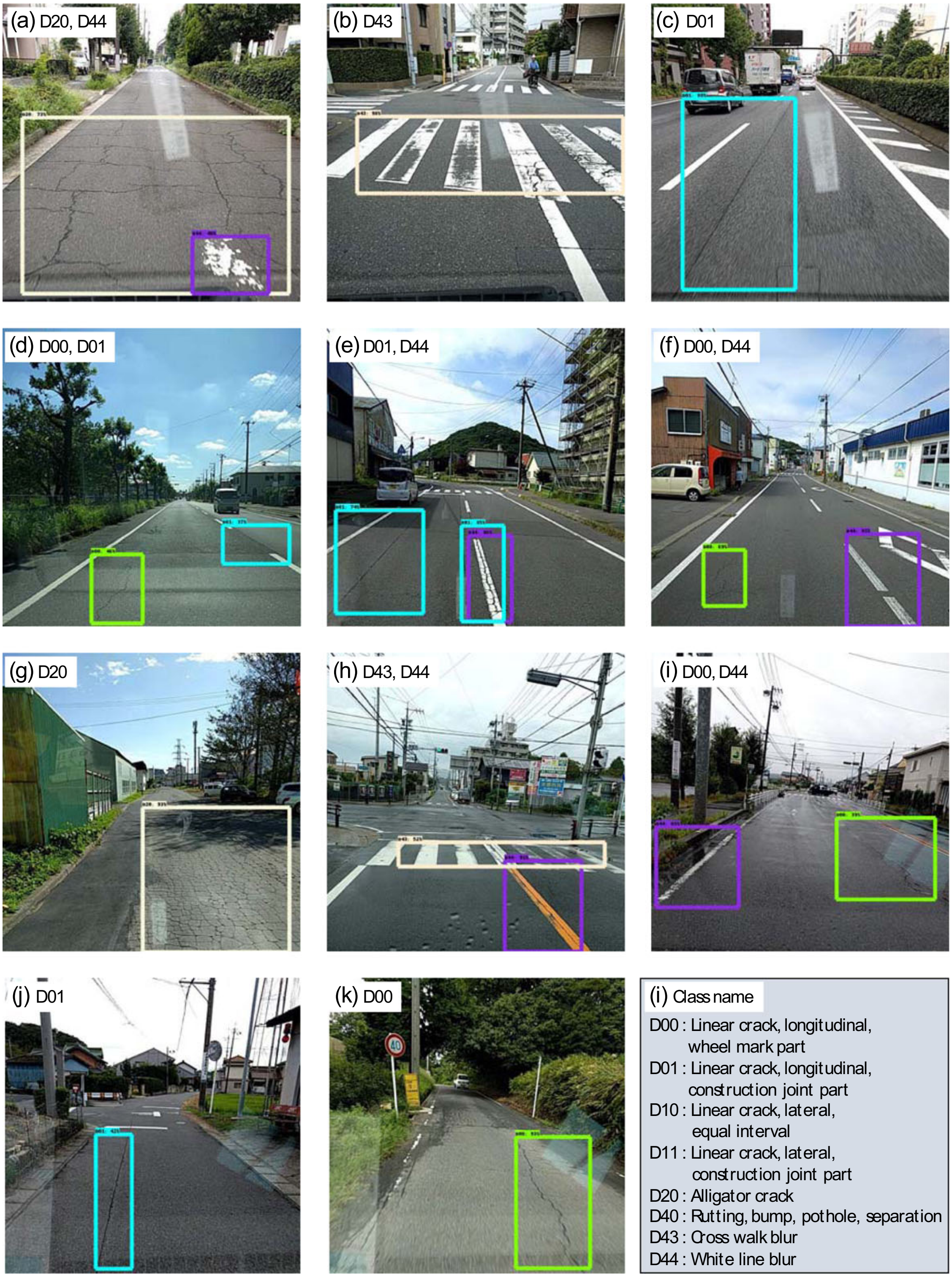

(a) D20, D44
(b) D43
(c) D01
(d) D00, D01
(e) D01, D44
(f) D00, D44
(g) D20
(h) D43, D44
(i) D00, D44
(j) D01
(k) D00
(i) Class name
D00 : Linear crack, longitudinal, wheel mark part
D01 : Linear crack, longitudinal, construction joint part
D10 : Linear crack, lateral, equal interval
D11 : Linear crack, lateral, construction joint part
D20 : Alligator crack
D40 : Rutting, bump, pothole, separation
D43 : Cross walk blur
D44 : White line blur

Fig. 5. Detected samples using the SSD MobileNet.

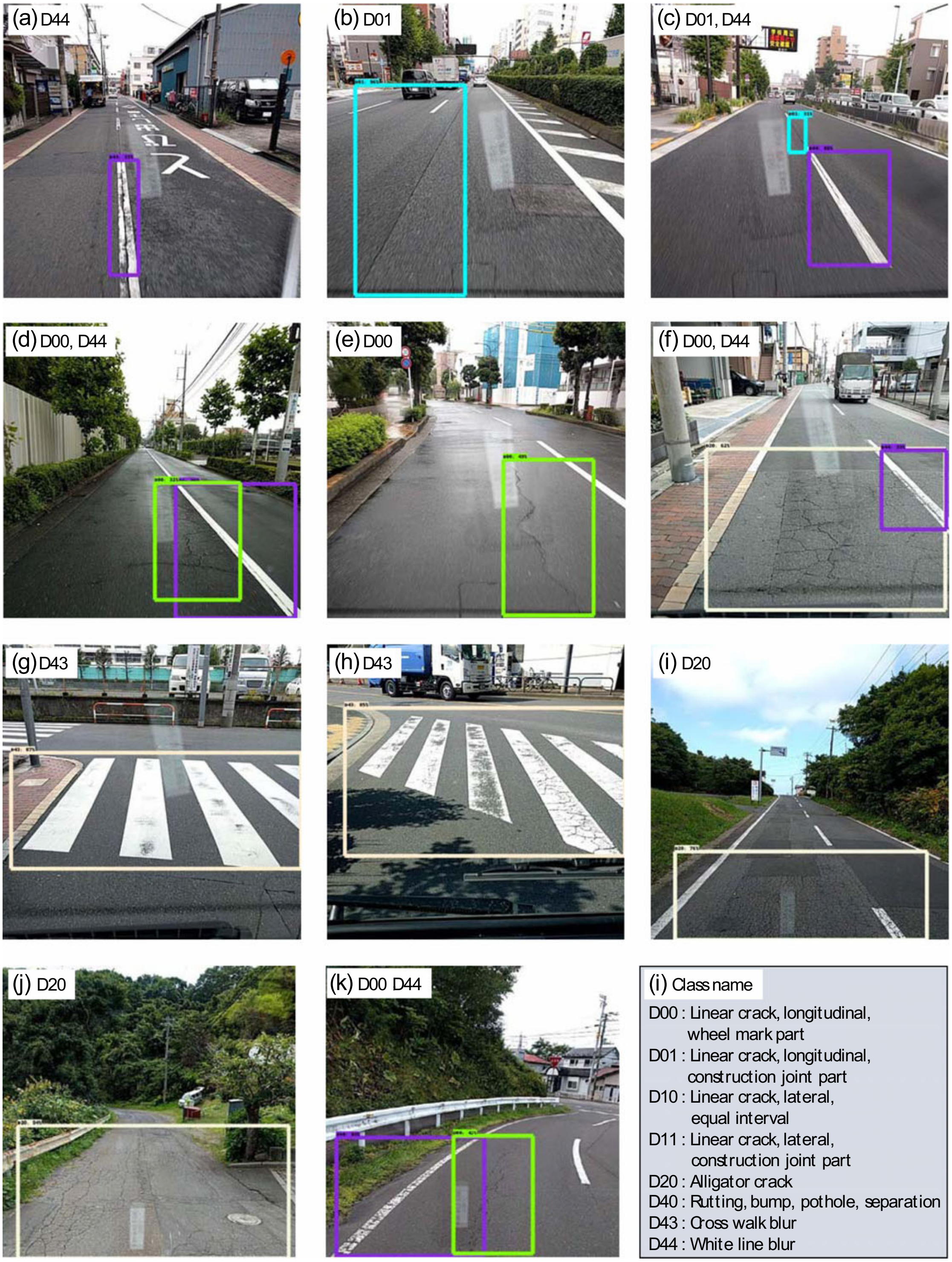

(a) D44
(b) D01
(c) D01, D44
(d) D00, D44
(e) D00
(f) D00, D44
(g) D43
(h) D43
(i) D20
(j) D20
(k) D00 D44
(i) Class name
D00 : Linear crack, longitudinal, wheel mark part
D01 : Linear crack, longitudinal, construction joint part
D10 : Linear crack, lateral, equal interval
D11 : Linear crack, lateral, construction joint part
D20 : Alligator crack
D40 : Rutting, bump, pothole, separation
D43 : Cross walk blur
D44 : White line blur

Fig. 6. Detected samples using the SSD Inception V2.

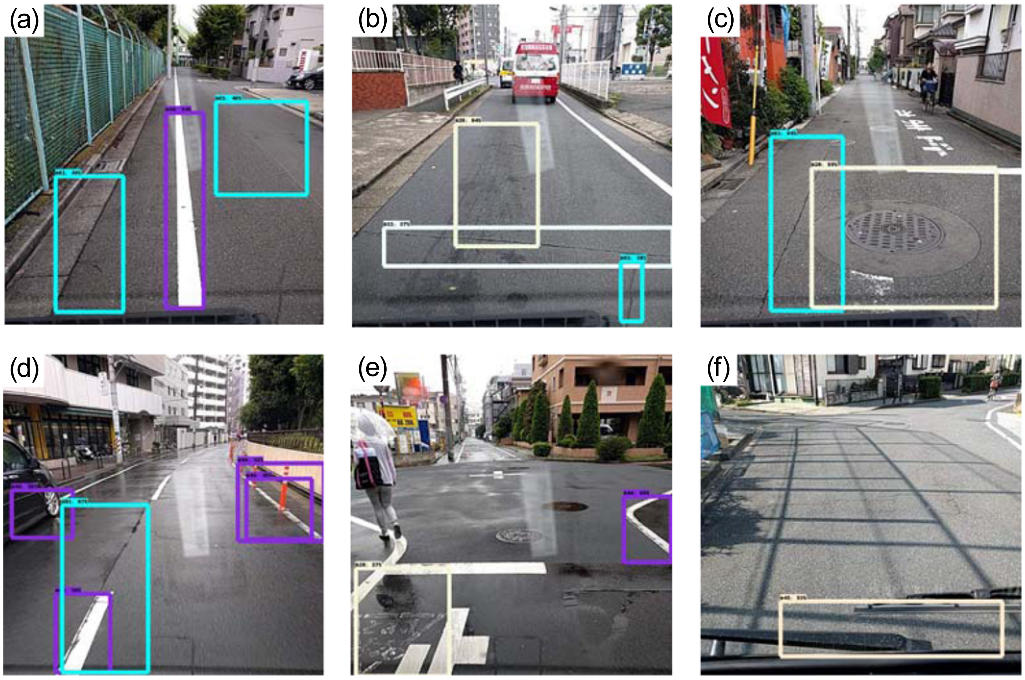


Fig. 7. Examples of false positives. Misjudgments of (a) a side groove as D01, (b) a wet road as D20, (c) a manhole as D20, (d) a car wheel as D44, (e) a streaked road sign as D20, and (f) a reflected wiper as D43.

Table 3
Inference speed (m/s) for each model for image resolution of a 300 × 300 pixel image

Model details	*Inference speed (m/s)*
SSD using Inception V2 (GPU)	63.1
SSD using MobileNet (GPU)	30.6
SSD using MobileNet (smartphone)	1,500

it is thought that this is why these misjudgments were made.

Moreover, typical false negative examples are shown in Figure 8. When some object was reflected on the windshield, damage was not detected correctly. Furthermore, when the blur of a pedestrian crossing was reflected horizontally, or when only a part of it was shown, it was not detected. Moreover, some potholes were not detected. This seems to be because there were few images where the damage was partially reflected in the training data.

Next, the inference speed of each model is described in Table 3. The speed was tested on a PC with the same specifications as in the previous case and a Nexus 5X smartphone with an MSM8992 CPU and 2 RAM GB memory. In this case, the SSD Inception V2 was two times slower than the SSD MobileNet, which is consistent with the result of Huang et al. (2017). In addition, because the smartphone processes data in 1.5 seconds, when it is installed in a moving car, damage to the road surface can be detected in real time and with the same accuracy as in Table 2. Our smartphone application, which we used to detect road damage using the trained model with our data set (SSD with MobileNet; Figure 9) is publicly available on our Web site. Please note that the damage detection and classification process is running on the smartphone.

6 DISCUSSION

To detect road damage accurately, it is important to obtain three-dimensional (3D) depth images. However, to acquire such images, a dedicated vehicle must be used. As such, it is not possible to inexpensively and exhaustively inspect all roads. Under such circumstances, we think that it is worthwhile to consider methods that can comprehensively survey road surfaces at low costs, such as methods that rely on smartphones. For example, it is possible to acquire data by attaching a smartphone to a parcel-delivery service, postal service, or public vehicle. Although the accuracy of the model obtained in this research is not high when compared to the proposals that use highly accurate sensors, the proposed model is nevertheless effective for a preliminary and exhaustive inspection of all roads in a district before more expensive

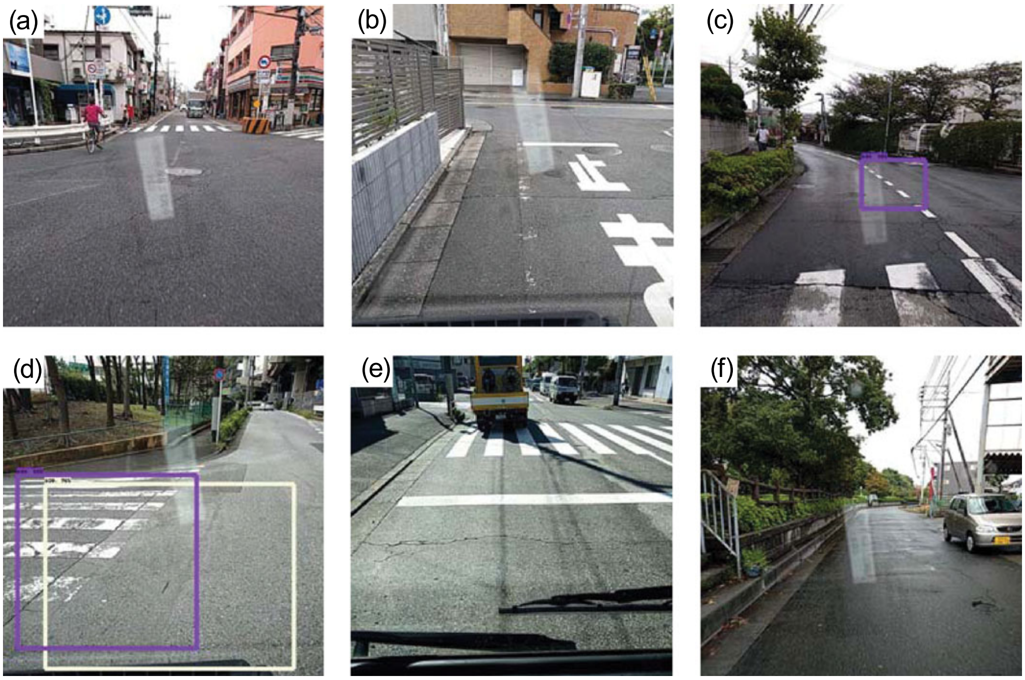


Fig. 8. Examples of false negatives. Images from (a) to (b) have the damages, respectively. (a) D00, (b) D44, (c) D43, (d) D43, (e) D10, and (f) D40 cannot be detected.

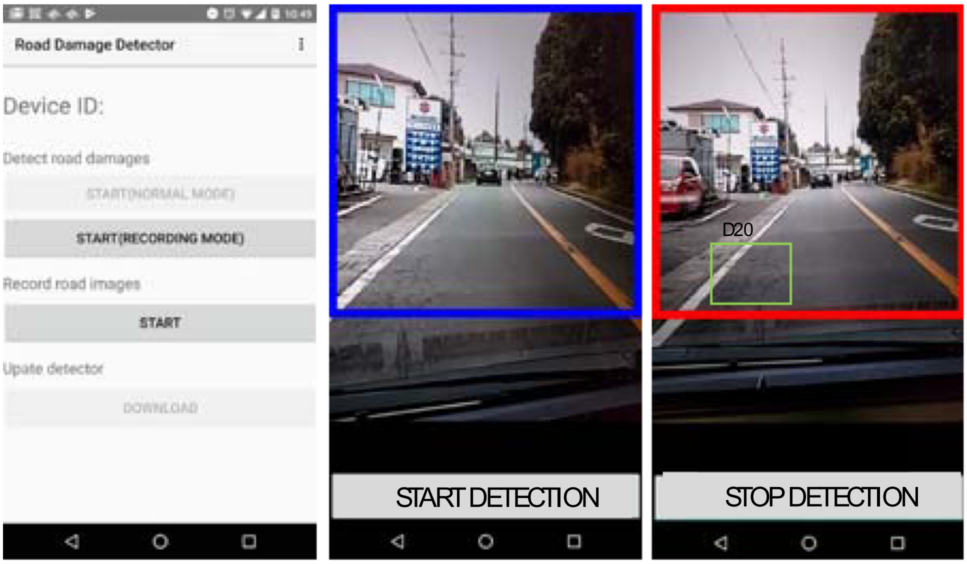


Fig. 9. Operating screen of our smartphone application. It is designed to be mounted on the dashboard of a general passenger car (see Figure 1). Detection of road surface damage is initiated by pressing the "START DETECTION" button. An image of the damaged part and the position information are transmitted to the external server only when damage is found. Using the SSD with MobileNet, this application can detect eight types of road damages within 1.5 seconds with the same accuracy as shown in Table 2.

methods are implemented. In other words, those roads that will require 3D depth images can be identified through preliminary inspections with a smartphone.

Considering that images on a smartphone can be processed every 1.5 seconds and that the road 10 m ahead is reflected by the in-vehicle smartphone, when vehicles travel at less than 6.6 m/s (23 km/h), we can completely inspect the road surface without any missing information. Even when traveling at faster speeds, public vehicles that travel regularly along the same route can perform comprehensive inspections over time. In fact, by installing a GPS to measure the routes of public vehicles over the course of a year, it was shown that public vehicles travel along over 80% of the roads in a municipality (Obara et al., 2017). Therefore, it is possible to survey roads comprehensively and at low cost by installing smartphones on public vehicles over an ample period of time.

In future research, we shall increase the amount of training data and devise the structure of a new neural network to improve the detection accuracy of categories that cannot be detected well. In general, object detection by deep learning requires more than 1,000 images for each class. By continuing our experiment over the long term in the future, we plan to increase the number of images that can be used as training data. We also believe that simulated training images can be effective, by using GAN (Generative adversarial network; Radford et al., 2015) and other such approaches. Further, although we here developed a benchmark by combining SSD, MobileNet, and Inception V2, it will also be helpful to devise a model that is more suitable for this data set. Further, a pixel-by-pixel-based method can be applied to detect road damages. Moreover, instead of solving the damage detection as a problem of image processing, the determination accuracy can be improved by combining with different kinds of data (e.g., vibration data). On the other hand, we believe we need to reconsider the definition of the damage types. In this article, although the definition was based on visual inspection standards in Japan's road management, some categories are too ambiguous to distinguish using images captured by a smartphone installed on the dashboard. For example, it is also important to devise other damage definitions, such as classifying damage types in descending order of risk after discovery. Finally, we shall determine whether a trained model can be used on images collected from locations other than Japan.

7 CONCLUSIONS

In this study, we developed a new large-scale data set for road damage detection and classification. In collaboration with seven local governments in Japan, we collected 163,664 road images. Then, these images with road damage were visually confirmed and classified into eight classes; out of these, 9,053 images were annotated and released as a training data set. To the best of our knowledge, this data set is the first one for road damage detection. We strongly believe this data set provides a new avenue for road damage detection. In addition, we trained and evaluated the damage detection model using our data set. Based on the results, in the best-detectable category, we achieved recalls and precisions greater than 71% and 77% using MobileNet and Inception V2, respectively, with an inference time of 1.5 seconds on a smartphone. We believe that a simple road inspection method using only a smartphone will be useful in regions where experts and financial resources are lacking. To support research in this field, we have made the data set, trained models, source code, and smartphone application publicly available.

ACKNOWLEDGMENTS

The research results have been achieved by "Development of next-generation citizen collaborative platform that organically incorporates local knowledge and citizen's knowledge by researches and developments for utilization and platforms of social big data," the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan. Additionally, we would like to express our gratitude to Ichihara city, Chiba city, Sumida ward, Nagakute city, Adachi ward, Muroran city, and Numazu city for their cooperation with the experiment.

REFERENCES

AASHTO (2008), *Bridging the Gap–Restoring and Rebuilding the Nation's Bridges*, American Association of State Highway and Transportation Officials, Washington DC.

Adeli, H. (2001), Neural networks in civil engineering: 1989–2000, *Computer-Aided Civil and Infrastructure Engineering*, **16**(2), 126–42.

Akarsu, B., Karaköse, M., Parlak, K., Erhan, A. K. I. N. & Sarimaden, A. (2016), A fast and adaptive road defect detection approach using computer vision with real time implementation, *International Journal of Applied Mathematics, Electronics and Computers*, **4**(Special Issue-1), 290–95.

Buttlar, W. G. & Islam, M. S. (2014), *Integration of Smart-Phone-Based Pavement Roughness Data Collection Tool with Asset Management System*, Technical Report, USDOT Region V Regional University Transportation Center, NEXTRANS Center, West Lafayette, IN.

Casas-Avellaneda, D. A. & López-Parra, J. F. (2016), Detection and localization of potholes in roadways using smartphones, *Dyna*, **83**(195), 156–62.

Cha, Y.-J., Choi, W. & Buyukozturk, O. (2017), Deep learning-based crack damage detection using convolutional neural networks, *Computer-Aided Civil and Infrastructure Engineering*, **32**(5), 361–78.

Chen, F. C., Jahanshahi, M. R., Wu, R. T. & Joffe, C. (2017), A texture-based video processing methodology using Bayesian data fusion for autonomous crack detection on metallic surfaces, *Computer-Aided Civil and Infrastructure Engineering*, **32**(4), 271–87.

Cheng, M. M., Zhang, Z., Lin, W. Y. & Torr, P. (2014), BING: binarized normed gradients for objectness estimation at

300fps, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ohio, 3286–93.

Chun, P.-J., Hashimoto, K., Kataoka, N., Kuramoto, N. & Ohga, M. (2015), Asphalt pavement crack detection using image processing and naïve Bayes based machine learning approach, *Journal of Japan Society of Civil Engineers, Ser. E1 (Pavement Engineering)*, **70**(3), 1–8.

Dai, J., Li, Y., He, K. & Sun, J. (2016), R-FCN: object detection via region-based fully convolutional networks, in *Proceedings of the Neural Information Processing Systems Conference*, Barcelona, Spain, 379–87.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), ImageNet: a large-scale hierarchical image database, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Florida, 248–55.

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stöckert, U. & Groß, H-M. (2017), How to get pavement distress detection ready for deep learning? A systematic approach, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2039–47.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2015), The Pascal visual object classes challenge: a retrospective, *International Journal of Computer Vision*, **111**(1), 98–136.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010), The Pascal visual object classes (VOC) challenge, *International Journal of Computer Vision*, **88**(2), 303–38.

Fan, Z., Wu, Y., Lu, J. & Li, W. (2018), Automatic pavement crack detection based on structured prediction with the convolutional neural network, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1802.02208.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. (2010), Object detection with discriminatively trained part-based models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9), 1627–45.

Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. (2013), Vision meets robotics: the KITTI dataset, *The International Journal of Robotics Research*, **32**(11), 1231–37.

Girshick, R. (2015), *Fast R-CNN*, in *Proceedings of the IEEE International Conference on Computer Vision*, Boston, MA, 1440–48.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014), Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ohio, 580–87.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Nevada, 770–78.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017), MobileNets: efficient convolutional neural networks for mobile vision applications, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1704.04861.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. & Murphy, K. (2017), Speed/accuracy trade-offs for modern convolutional object detectors, in *Proceedings of the IEEE CVPR*, Honolulu, HI.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F., Coates, A. & Ng, A. Y. (2015), An empirical evaluation of deep learning on highway driving, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1504.01716.

Ioffe, S. & Szegedy, C. (2015), Batch normalization: accelerating deep network training by reducing internal covariate shift, in *Proceedings of the International Conference on Machine Learning*, Lille, France, 448–56.

Jo, Y. & Ryu, S. (2015), Pothole detection system using a black-box camera, *Sensors*, **15**(11), 29316–31.

JRA (2013), *Maintenance and Repair Guide Book of the Pavement 2013*, 1st edn., Japan Road Association, Tokyo, Japan.

Kawano, M., Mikami, K., Yokoyama, S., Yonezawa, T. & Nakazawa, J. (2017), Road marking blur detection with drive recorder, in *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, 4092–97.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014), Microsoft COCO: common objects in context, in *Proceedings of the European Conference on Computer Vision*, Springer, Berlin, 740–55.

Lin, Y. Z., Nie, Z. H. & Ma, H. W. (2017), Structural damage detection with automatic feature-extraction through deep learning, *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1025–46.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A. C. (2016), SSD: single shot multibox detector, in *Proceedings of the European Conference on Computer Vision*, Springer, Berlin, 21–37.

Maeda, H., Sekimoto, Y. & Seto, T. (2016), Lightweight road manager: smartphone-based automatic determination of road damage status by deep neural network, in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, Burlingame, CA, 37–45.

Mertz, C., Varadharajan, S., Jose, S., Sharma, K., Wander, L. & Wang, J. (2014), City-wide road distress monitoring with smartphones, in *Proceedings of ITS World Congress*, Tokyo, Japan, 1–9.

Ministry of Land, Infrastructure, Transport and Tourism (2016), White Paper on Present state and future of social capital aging. Infrastructure maintenance information, 2016 (in Japanese).

Mohan, A. & Poobal, S. (2017), Crack detection using image processing: a critical review and analysis, *Alexandria Engineering Journal*, https://doi.org/10.1016/j.aej.2017.01.020.

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.

Obara, M., Kashiyama, T., Sekimoto, Y. & Omata, H. (2017), Analysis of public vehicle use with long-term GPS data and the possibility of use optimization through working car project, in *Proceedings of The Third International Conference on Smart Portable, Wearable, Implantable and Disability-Oriented Devices and Systems (SPWID 2017)*, Venice, Italy.

O'Byrne, M., Ghosh, B., Schoefs, F. & Pakrashi, V. (2014), Regionally enhanced multiphase segmentation technique for damaged surfaces, *Computer-Aided Civil and Infrastructure Engineering*, **29**(9), 644–58.

Radford, A., Metz, L. & Chintala, S. (2015), Unsupervised representation learning with deep convolutional generative adversarial networks, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1511.06434.

Rafiei, M. H. & Adeli, H. (2017), A novel machine learning-based algorithm to detect damage in high-rise building structures, *The Structural Design of Tall and Special Buildings*, **26**(18), https://doi.org/10.1002/tal.1400.

Rafiei, M. H. & Adeli, H. (2018), A novel unsupervised deep learning model for global and local health condition assessment of structures, *Engineering Structures*, **156**, 598–607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R. & Adeli, H. (2017), Supervised deep restricted Boltzmann machine for estimation of concrete, *ACI Materials Journal*, **114**(2), 237–44.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016), You only look once: unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Nevada, 779–88.

Redmon, J. & Farhadi, A. (2017), YOLO9000: better, faster, stronger, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI.

Ren, S., He, K., Girshick, R. & Sun, J. (2015), Faster R-CNN: towards real-time object detection with region proposal networks, in *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, 91–99.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y. (2013), OverFeat: integrated recognition, localization and detection using convolutional networks, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1312.6229.

Simonyan, K. & Zisserman, A. (2014), Very deep convolutional networks for large-scale image recognition, *Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1409.1556.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016), Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Nevada, 2818–26.

Tomiyama, K., Kawamura, A., Fujita, S. & Ishida, T. (2013), An effective surface inspection method of urban roads according to the pavement management situation of local governments, *Journal of Japan Society of Civil Engineers, Ser. F3 (Civil Engineering Informatics)*, **69**(2), I-54–I-62.

Uijlings, J. R., Van De Sande, K. E., Gevers, T. & Smeulders, A. W. (2013), Selective search for object recognition, *International Journal of Computer Vision*, **104**(2), 154–71.

Zalama, E., Gómez-García-Bermejo, J., Medina, R. & Llamas, J. (2014), Road crack detection using visual features extracted by Gabor filters, *Computer-Aided Civil and Infrastructure Engineering*, **29**(5), 342–58.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J. Q. & Chen, C. (2017), Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network, *Computer-Aided Civil and Infrastructure Engineering*, **32**(10), 805–19.

Zhang, K., Cheng, H. D. & Zhang, B. (2018), Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning, *Journal of Computing in Civil Engineering*, **32**(2), 1–12.

Zhang, L., Yang, F., Zhang, Y. D. & Zhu, Y. J. (2016), Road crack detection using deep convolutional neural network, in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, 3708–12.