

# Concrete Crack Detection Using UAV and Deep Learning

Saki Murao

*Graduate Student, Dept. of Civil and Earth Resource Engineering, Kyoto University, Kyoto, Japan*

Yasutoshi Nomura

*Lecturer, Dept. of Civil Engineering, Ritsumeikan University, Kusatsu, Japan*

Hitoshi Furuta

*Professor, Faculty of Informatics, Kansai University, Takatsuki, Japan*

Chul-Woo Kim

*Professor, Dept. of Civil and Earth Resource Engineering, Kyoto University, Kyoto, Japan*

**ABSTRACT:** In recent years, the number of infrastructures to be inspected or repaired has increased, because many of them are suffering from the degradation and deterioration. However, it is difficult to acquire the sufficient number of experienced engineers due to their aging and retirement. Usually, when judging the degree of damage of the infrastructure, a visual inspection is carried out by experienced engineers as a preliminary checking for non-destructive examination. Recently, it is expected to introduce robots such as Unmanned Aerial Vehicles (UAV) for the inspection because it can reduce the burden on engineers and collect data from the spaces where people are difficult to approach.

However, it is often difficult to secure sufficient accuracy for the data collected by robots. This problem can be solved by adjusting the photographing angle and distance if possible to confirm the detection accuracy of the crack contained in the photographed image on the spot. Then, robotic inspection is available as a useful tool that can realize the real-time detection.

In this research, an attempt is made to develop an efficient crack detection system for concrete bridge structure, using You Only Look Once (YOLO) method that is a method possessing the possibility of real-time processing. In order to demonstrate the applicability of the system with YOLO, an experiment was conducted by applying the system to the crack image of concrete wall. From the experiments, it was found that the detection time was highly small and further detection was possible even in real-time. Regarding the accuracy, although it is possible to identify the location where cracks occurred roughly, it is difficult to detect perfectly, and erroneous detection was also observed. For these reasons, further improvement in accuracy is necessary for practical use.

## 1. INTRODUCTIONS

### 1.1. Current status of bridge inspection

In recent years, the number of bridges requiring repair and rebuilding has increased because of their deterioration and degradation of existing. In Japan, many bridges were built during the period of high economic growth. The number of bridges built by 1970 is about 41,000. Accordingly, in the

near future, the number of bridges over 50 years from construction will increase rapidly. In addition, the traffic volume nowadays is twice in the past twenty years. For these reasons, regular inspections of road bridges are obliged to conduct through visual inspection every five years (Ministry of Land, Infrastructure and Transport 2015). However, most of the members of the bridges are located in high places and difficult to

approach. For the inspection such structural members, a lot of labors and cost are required. Namely, it is necessary for the inspection to install scaffolds and use special vehicles or equipment.

Meanwhile, there is another problem that local governments cannot fully perform the maintenance work due to the lack of enough budget and skilled maintenance engineers by the economic slackening and retirement.

### *1.2. Improvement of bridge inspection efficiency*

Various researches have attempted to improve the efficiency of bridge inspection. As an alternative for inspection work, a number of health monitoring technologies have been developed, which attach sensors to bridges to detect damage or abnormal phenomena. (Fujino 2013, Ogawa, et al. 2009). However, it is difficult to implement monitoring for all infrastructures because of the durability of the sensor and the secure power supply.

In recent years, a bridge inspection robot such as Unmanned Aerial Vehicles (UAV) has attracted attention to reduce cost and improve the efficiency of inspection work (Okajima 2016). For example, using a UAV equipped with a camera, it is possible to obtain the photos of the surface condition of a bridge without scaffold for inspection. However, there remain several problems to be solved for practical use: the dropping of the robot, the insufficient quality of the photo images obtained from the motion pictures, and so on. This study aims to establish a system to detect damages such as cracks and corrosion with ease from the limited duration for the operation of the UAV. Furthermore, this study investigates the issues for practical application of the inspection robots, and examines the applicability of image recognition technology as a solution to those problems.

### *1.3. Automatic detection of damage by image recognition*

Image recognition is a technique for automatically recognizing or identifying what is shown in an image. With the use of deep learning, drastic improvements in the accuracy of image

recognition can be achieved in various fields including general object detection such as face authentication. It has also been developed in the field of infrastructure engineering (Yokoyama, Matsumoto 2016, 2017). In this research, a new system is proposed to detect cracks, using UAV and image recognition technology.

Before deep learning appeared, image feature quantities are extracted from differences in luminance values, images are recognized by comparing feature values of images to be judged by grouping points having feature values close to each other. In this method, there is a problem that the image feature amount changes depending on the appearance and the angle, where the positional relationship is ignored in the structure of the data to be handled, which is affected by the variations in density of image feature points (Takei 2016).

In image recognition using deep learning, feature extraction is performed automatically, and since the image itself is used as input data, preliminary processing such as acquiring image feature points from the image is not necessary. In other words, features can be learned automatically without extracting and selecting characteristic features themselves. Due to the improvement of recognition accuracy, the application of image recognition has attracted a great deal of attention (Yokoyama, Matsumoto 2016, 2017). A method to learn images of cracks and rusts of infrastructures, especially for concrete structures, was developed to automatically detect damage and structural deformation from the images acquired at the time of inspection by the deep learning. The method is based on the general object detection technology of deep learning, however it takes a few seconds to detect the damage from an image when using conventional deep learning techniques. Then, it may take a lot of time to inspect a large structure.

When performing inspection using a robot with the conventional system which takes time to detect damage as described above, the processes of taking pictures by the robot and detection processing are performed independently. Since bridge inspection is susceptible to environmental

influences such as backlighting and wind, it is difficult to secure the quality of images capturing damaged parts, and there is a possibility that the collected data cannot be used for the detection. Therefore, even if a system with high detection accuracy is constructed, the system is not available until the suitable data for detection is collected.

In this research, it aims to develop a new system using a robot to detect damage in real-time. If the real-time damage detection is realized, the accuracy of the collected data is checked during the operation and examine the necessity of more detailed data collection of the detection site. This means that it is possible to enhance the usefulness of the data. Then, it is noted that there is a trade-off relation between detection accuracy and calculation time. Therefore, it is attempted here to construct a new system that can detect damage in real-time from the viewpoints of calculation speed and detection accuracy.

## 2. METHODOLOGY

YOLO (Redmon and Farhadi 2016) is provided as a function of the framework “Darknet” coded in C (Fujita and Takahara 2016). YOLO can estimate the position, size, and type of an object in an image including plural different kinds of objects. A major feature of YOLO with regard to object detection is that extraction of a candidate region of a detection target object and calculation of class probability of the candidate region are performed with one estimate at the same time as shown in Figure 1.

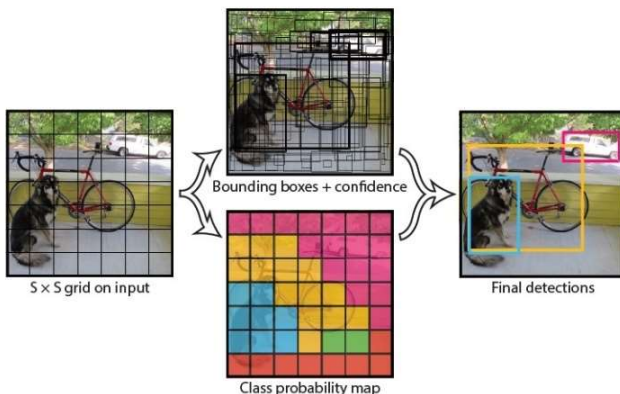


Figure 1: Object detection procedure of YOLO  
(Redmon et al. 2016).

For this reason, this method makes it possible both to perform object detection at a very high speed, and to perform the processing at about 1000 times faster than the conventional method. Therefore, it can be considered that the method is useful to achieve the real-time detection.

## 3. CASE STUDY FOR CRACK DETECTION

In this chapter, the deep learning described in Chapter 3 is applied to detect crack damage on concrete slabs to verify the usefulness of the system.

### 3.1. System overview

In the proposed system, an attempt is made to extract cracks from images of concrete wall taken with a digital video camera equipped on an UAV. Since our objective is to construct a system that can detect cracks in real-time, YOLOv2 (Redmon and Farhadi 2016), which is an improved system of YOLO is used, because it has higher accuracy and higher speed processing among general object detection algorithms using deep learning.

For training data, various kinds of images of cracks (i.e. not limited to cracks in concrete) were used, and information on cracked areas and positions in each image was extracted as training data. Then, test images were applied to the system for crack detection.

### 3.2. System construction

In this research, three systems (Systems 1 to 3) were constructed while changing training data. For the training data, we used images captured using digital cameras and various image data collected from the Internet. From these images, a cracked region was cut out, and the positional information of the object, such as the coordinates and the size of the region, were created. BBox-Label-Tool was used to obtain object position information from image data. BBox-Label-Tool is a range selection tool for images implementing in Python (Fujita, Takahara 2016). This tool reads images and provides coordinate data in text data

by enclosing the object with a rectangular frame. This position information is read inside YOLOv2, and training data is generated.

In System 1, two classes were prepared according to the object to be detected: “crack” which is related to crack class and “chalk” related to crack traced with chalk. The training data was created from 60 images of “crack” and 104 images of “chalk” and run through the program 40,000 times. After creating position information of cracks, System 1 enabled to identify cracks without fail (Figure 2).

Inside YOLOv2, training images are randomly resized, inverted, rotated, HSV (Hue, Saturation, Value) model converted to increase is the number of different types of images.



Figure 2: Example of training data of System 1 (crack class)

System 2 is used to identify concrete joints with line forms as other than cracks, which is named “etc” class. Then, the training data for “crack” is set as that the frame of position information is larger than it in System 1 in order to learn the shape of the crack clearer. In addition, branch-crack class were set for the purpose of detecting crack shape and making feature extraction easier. Examples of the “etc”, “crack” and “branch-crack” classes are respectively shown in Figures 3, 4 and 5.

The training data of each class was created from 31 images of “crack”, 23 image of “chalk”, 33 images of “branch-crack”, 27 images of “branch-chalk”, and 52 images of “etc”, by running 40,000 times. When multiple classes are

included in one image, the same image is used for learning each class.



Figure 3: Example of the etc class in the training data for System 2



Figure 4: Example of the crack class in the training data for System 2



Figure 5: Example of the branch-crack class in the training data for System 2

In System 3, photographs of a wall at Kansai University, which are similar as the images used for the test, are added to the training data of System 2. However, the training data added System 3 are different from the test data. The settings for classes and training data were the



same as ones of System 2. For each class, learning was implemented 40,000 times using training data created from 54 images of “crack”, 51 images of “chalk”, 28 images of “branch-crack”, 68 images of “branch-chalk”, and 52 images of “etc”.

In System 2 and System 3, it was confirmed that the training data on the “chalk class” is different, but the recognition rate of crack is not affected. Therefore, we will focus on the detection of cracks and will verify the applicability of the proposed system.

In all of these systems, we used a GeForce GTX 1080 Ti Desktop computer with 64 GB of RAM and 4.20 GHz of CPU speed.

### 3.3. Application results

For the verification of the constructed Systems 1 to 3, 21 images of concrete walls located at the campus of Kansai university were prepared as test images and applied to each system.

An example of test data and its correct detection are shown in Figure 6. Detection results for test data of each system with the representative images are listed in Table1.

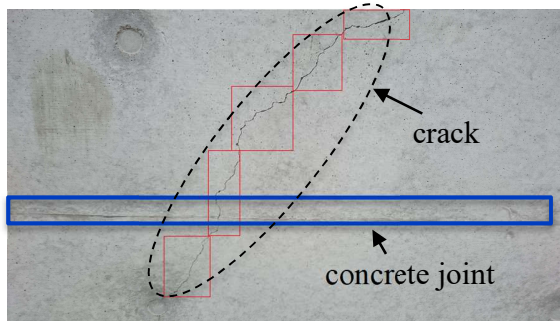


Figure 6: Example of correct detection

Table 1: Detection examples

System	Detection example
System 1	

System 2	
System 3	

The computation time to detect cracks was about 0.1 seconds, which is considered to be able to detect cracks in real-time.

Although System 1 generally detects cracks well, it misses partial cracks in many cases and moreover often cannot detect one crack completely. In addition, it showed some erroneous detection where straight lines such as concrete joints were detected as shown in Table1. That is because finer crack regions were employed in the training data so that a lot of concrete joint line data was learned as “crack”.

In System 2, the incorrect detection of region different from crack region has been greatly reduced as shown in Figure 7. However, the ability of detection of cracks became remarkably worse as shown in Table1. In addition, erroneous detection such as detection of region with different from crack was also seen as shown in Figure 7. The reason for this is considered that the number of training data decreased due to re-setting the frame larger to cut out so as to make it easier to extract the characteristics of cracks when preparing training data.

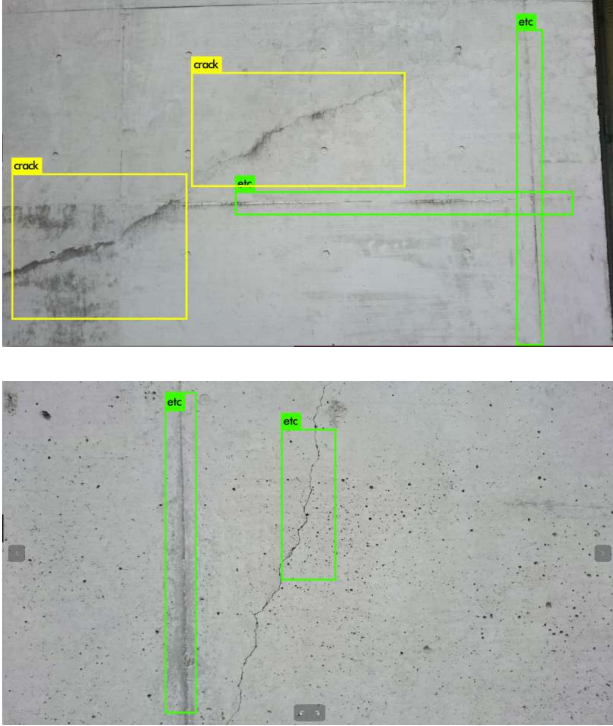


Figure 7: Detection examples of System 2

Although System 3 is able to detect cracks completely, it appears to be better to detect cracks than System 2 as shown in Table 1. Even for the images where cracks could not be detected at all in System 2, System 3 was able to detect them, although the detection is not perfect. However, similar to System 2, it was confirmed that items other than cracks were judged as cracks.

### 3.4. Comparison of accuracy between System 2 and System 3

Based on the application results for 21 test images, the precision ratio and the recall ratio in System 2 and System 3 were calculated and compared.

For the calculation method, frames of the correct answers were determined for test images in the same way as the training data. Examples of the frames of the correct answer are shown in Figure 8. Then, the correct answers are compared to the detected results for 21 images respectively. Figure 9 shows an example of the detection result. This study evaluates precision ratio and recall ratio to compare the correct answer and the detection results.

The precision ratio indicates how often correctly classified answers are obtained, and the recall ratio indicates how many correct answers are detected. The precision ratio and the recall ratio were respectively assessed by the following equations.

$$\text{precision ratio} = tp / (tp + fp) \quad (1)$$

$$\text{recall ratio} = tp / (tp + fn) \quad (2)$$

where  $tp$ ,  $fp$  and  $fn$  respectively stand for the sample number of the *true positive*, *false positive* and *false negative* samples. That is,  $tp$  stands for the correctly detected samples,  $fp$  stands for the incorrectly detected samples, and  $fn$  stands for the correct answers that were not detected. The average values of the precision ratio and recall ratio calculated from the 21 images are compared in Table 2. It is noted that calculation is performed only for cracked regions, and the accuracy of detection for the “etc class” is not calculated.

The results summarized in Table 2 show that both of the precision and recall ratios are improved in System 3. In particular, recall ratio has greatly increased. The distribution of recall ratios is shown in Figure 10, which also demonstrates that System 3 detects cracks more accurately than System 2. From this result, it is conceivable that the detection accuracy increases as the number of training data increases, and as the training data includes samples similar to the test data. However, the detection accuracy of some images was not improved. No feature common to these images was found, nor could found any relation to learning data. In the future, it is also necessary to verify whether improvements can be achieved by increasing the number of training data, or by changing the method of taking pictures (i.e. angle, brightness etc). Although System 3 improves the accuracy of crack detection, it is still a challenging task to obtain a perfect trace of cracks, thus there is a room for improvement.

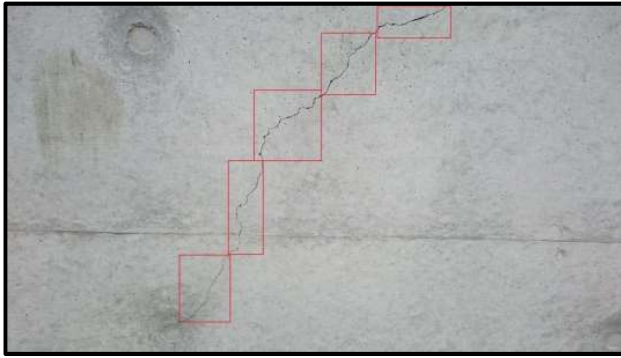


Figure 8: Example of correct answer marks

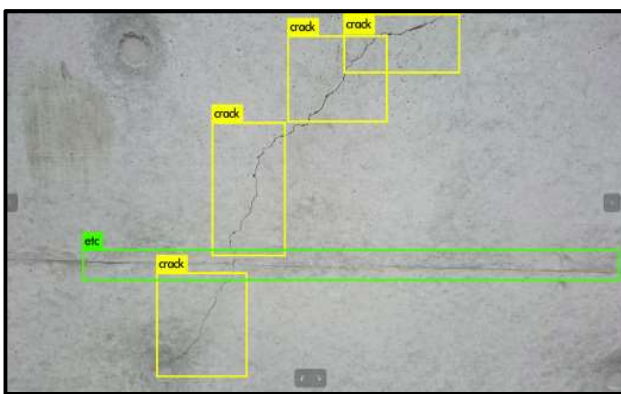


Figure 9: Example of the detection result

Table 2: Comparison of accuracy

System	Precision ratio	Recall ratio
System 2	27.95 %	27.96 %
System 3	41.63 %	60.75 %

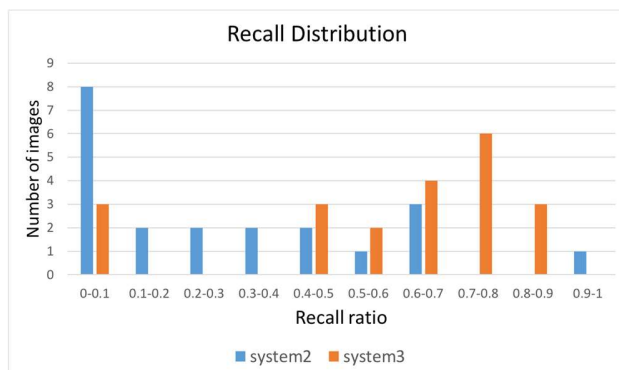


Figure 10: The distribution of recall ratios

#### 4. CONCLUSIONS

In this study, for the purpose of improving the efficiency of bridge inspection, the applicability of an UAV equipped with a camera for practical inspection use was investigated. Real-time detection of damaged parts in collected image data plays an important role in use of UAV. Learning Systems were constructed by using YOLO, which is one of deep learning methods capable of efficient object detection. The proposed system can detect cracks in real-time from images taken by the UAV.

Conclusions derived from this study are as follows.

- The computation time to detect cracks was extremely short. It can be said that this method is practically effective. It was observed that the calculation time was comparable to existing methods. It is noted that accuracy cannot be ensured unless the image is of sufficient resolution required to detect cracks.
- Concerning detection accuracy, it was possible to identify locations of cracks. However, the detection accuracy of cracks was about 60%. This means that further consideration is required for the improvement of detection accuracy.
- Comparative studies on the influence of number of training data showed that detection accuracy might be improved as the number of learned data increases. Therefore, in order to improve accuracy, it is required to acquire sufficient high-quality training data.
- The proposed approach is possibly applicable as a method of detection for other forms of damage such as corrosion, cracks, exposed steel bars and so on.

Further improvement of accuracy is necessary for practical use of the system, and for that purpose it is necessary to obtain many data. However, it is currently difficult to obtain data from actual structures. In the future, there is a possibility that accuracy can be improved by using automatic image generation technology (Saito 2016) using deep learning and supplementing/securing less training data. Then,

it is necessary to verify not only the accuracy of the system, but also crack that improvements can be made with regard to the image data acquisition.

Furthermore, if accuracy improves, it is expected that the proposed system will contribute to ensure the safety of a wide range of social infrastructures through faster, safer, and more efficient infrastructural damage detection for practice.

## 5. REFERENCES

- Ministry of Land, Infrastructure and Transport (2015). "Field verification and evaluation results of bridge maintenance and management", <http://www.mlit.go.jp/common/001083014.pdf#search=%27> (in Japanese) (Feb.5, 2019)
- Okajima, M. (2016). "Collection of monitoring data on bridges using compact drone", *Graduation Thesis, Faculty of Informatics, Kansai University*. (in Japanese)
- Yokoyama, M. and Matsumoto, T. (2016). "Development of automatic crack detector for concrete using deep learning", *The 19th Symposium on Applied Mechanics, JSCE*, 125-126. (in Japanese)
- Redmon, J. and Farhadi, A. (2016). YOLO9000: "Better, Faster, Stronger", *Computer Vision and Pattern Recognition*, 1-9.
- Ministry of Land, Infrastructure and Transport (2015). "Infrastructure and Transport", <http://www.mlit.go.jp/road/sisaku/yobohozen/torikumi.pdf> (in Japanese) (Nov.8, 2018)
- Fujino, Y. (2013). "Issues and prospects of sensing and monitoring technology", *Railway Research Review (RRR)*, vol.70, No.11, 4-9
- Ogawa, S., Kameda, K., Sato, H., and Mita, A. (2009). "Research on practical application of structural health monitoring - Construction of infrastructure System including data model -", *Proceedings of the Japan Earthquake Engineering Association*, vol.9, No.5
- Yokoyama, M. and Matsumoto, T. (2017). "Development of automatic deformation detector of concrete by Deep Learning and implementation of Web System", *The 20th Symposium on Applied Mechanics, JSCE*, 31-32. (in Japanese)
- Takei, H. (2016). "Deep learning in the beginning", 58-67. (in Japanese)
- Darknet. "Open Source Neural Network in C", <https://pjreddie.com/darknet/> (in Japanese) (Nov.8, 2018)
- Fujita, K., Takahara, A. (2016). "Implemented deep learning", 46, 150-151, 212-218.
- Saito, Y. (2016) "Create from scratch Deep Learning - Learning with Python Deep learning theory and implementation-", 77-79, 267-268. (in Japanese)
- Nomura, Y., Murao, S., Sakaguchi, Y. and Furuta, H. (2018). "Crack detection System for concrete surface based on deep convolution neural network", *JSCE*, Vol.73, No.2. (in Japanese)