A thesis for the degree of Doctor of Philosophy

A study on deep learning-based automatic defect detection for social infrastructure maintenance

社会インフラの維持管理支援のための深層学習に基づく自動変状検出に関する研究

An Wang

Graduate School of Information Science and Technology

Hokkaido University

September, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With the growth of the global economy, various infrastructures such as tunnels, bridges, and viaducts have been constructed successively. These structures play pivotal roles in both economic development and transportation. In Japan, a significant portion of the infrastructure built during the rapid economic growth period is aging, and the number of inspectors is also decreasing due to labor shortages caused by the aging population [1]. The maintenance and management costs of Japan's infrastructure are expected to increase significantly after 20 years. Thus, reducing costs and improving efficiency in ensuring safety and managing the maintenance of existing infrastructure have become pressing concerns for the government.

In recent years, governments worldwide have put digitalization on the agenda, and so as digitalization of the infrastructure construction and maintenance field. To address the aforementioned situation, the Japanese government is taking action. In the revitalization strategy proposed by the Japanese government in 2016, which aimed to explore the application of new technologies, such as robots, across various fields, including infrastructure maintenance management, and to promote research, development, and verification of related technologies. Additionally, following the government strategies, the Ministry of Land, Infrastructure, Transport, and Tourism proposed the i-Construction to initiate the development and application of next-generation technologies, including the development and application of automated infrastructure maintenance management technologies.

Among various infrastructure components, aging bridges and tunnels are considered major

problems in transportation infrastructure. In particular, subway lines primarily consist of tunnels [2], and the window for inspecting these tunnels is brief, typically between the end of one work shift and the beginning of another. Therefore, a rapid and automated inspection method is required.

In research aimed at enhancing infrastructure maintenance and management, various methods have been proposed for automatic inspection. These methods employ conventional defect detection, robots, and other technologies [3–5]. However, even when these methods are implemented in inspection tasks, the final decisions regarding necessary countermeasures for inspected sites must be made by humans. There remains a need for technologies that directly assist engineers in this process. In the literature [2], one method for rapid automatic inspection involves detecting defects from images captured by visible light cameras. The paper presents a practical system that verifies cracks by photographing the inner surface of tunnels using a vehicle-mounted camera. Furthermore, in the literature [6,7], methods to detect cracks with high accuracy from images of the inner surface of tunnels were proposed. However, since various defects such as water leakage, junctions, and others appear in tunnels in addition to cracks, there is a need for automatic multi-scale defect detection methods.

Recently, the field of computer vision has seen significant improvements in image recognition performance due to the emergence of deep learning, which has proven useful in various tasks [7–11]. Therefore, it is anticipated that image recognition technology will facilitate the development of detectors capable of automatically identifying defects in infrastructure. Deep learning-based methods have outperformed traditional methods that use handcrafted image features to detect defects in infrastructure [12]. However, when developing deep learning methods to detect defects in infrastructure images, three main challenges persist: high-resolution image processing, multi-scale defect detection, and shortage of sub-pixel object detection capabilities issues.

- High-resolution image problem. In the first challenge, infrastructure defects often occupy small areas within the large image regions. This requires adapting the model to handle the foreground-background imbalance and developing a defect detection mechanism suitable for high-resolution images.

- Multi-scale defect detection problem. To address the second challenge, it is crucial to recognize that infrastructure can exhibit a variety of defects that differ in size, type, and subsequent implications for repair strategies. Therefore, simultaneous detection of these diverse defects is essential.

- Sub-pixel object detection problem. Cracks and other sub-pixel anomalies comprise a significant portion of infrastructure defects, often blended with similar structures in the background. Thus, it is imperative to enhance the sub-pixel object detection capability.

1.2 Research purpose

To address these issues, the contributions of this thesis are summarized as follows.

- This thesis explores the application of deep learning methods to high-resolution infrastructure imagery datasets.

- This thesis improves the semantic segmentation framework and optimizes the dataset to improve the ability of multi-scale defect detection and alleviate the problem of foreground and background imbalance in the data.

- This thesis improves the network's ability to detect sub-pixel anomalies while maintaining multi-scale detection capabilities. The proposed method improves the detection ability of the model in the infrastructure dataset and provides efficient spot-check support for practitioners in this field.

Specifically, I first develop a collaborative method using different deep-learning models, combining the strengths of fully convolutional network (FCN) and convolutional neural network (CNN) to capture defects across various scales. FCN, designed to be input-resolution-independent, can handle high-resolution images efficiently. Next, I further develop a defect detection approach using U-Net to reduce the computational load of previous methods to improve usability and merge the benefits of FCN and CNN with proven efficacy. The network structure was refined by introducing a defect detection technique using the atrous spatial pyramid pooling and inception modules to accommodate a wide array of defects. This effectively tackles issues such as background-foreground imbalance, multi-scale objects, and feature resemblance. By integrating these modules with the

U-Net architecture, the proposed method surpasses the traditional FCN-based techniques. Finally, to improve the detection accuracy of sub-pixel objects, I introduce the HRNet as the backbone and introduce an attention mechanism to enhance the ability of the network to detect sub-pixel objects.

The remainder of this thesis consists as follows. In Chapter 1, the background and objectives of this research are introduced. In Chapters 2 and 3, related works of semantic segmentation and the details of the dataset used in this research are respectively introduced. In Chapter 4, a CNN-based defect detection method is presented. Chapter 5 focuses on the combination of FCN and CNN methods for high-resolution subway tunnel images. In Chapter 6, the limitations of CNN and FCN in tunnel image defect detection are discussed, and a new U-Net-based defect detection method is proposed. In Chapter 7, I compare the constructed U-Net-based defect detection method with various semantic segmentation methods and confirm the effectiveness of the proposed approach in identifying issues such as long-tail problems and inadequate accuracy in detecting sub-pixel objects during the application process. In Chapter 8, an improved version of U-Net is proposed to enhance the capability of the defect detection method. In Chapter 9, a new HRNet-based network architecture is proposed to enhance the robustness of the defect detection method. Through these modifications, the efficacy of the proposed enhancement approach in improving the detection accuracy is confirmed. Finally, in Chapter 10, I provide a summary of this thesis, highlighting its contributions, limitations, and potential directions for future research. Figure 1.1 shows the relationships among the research ideas, strategies, and chapter structure in this study.

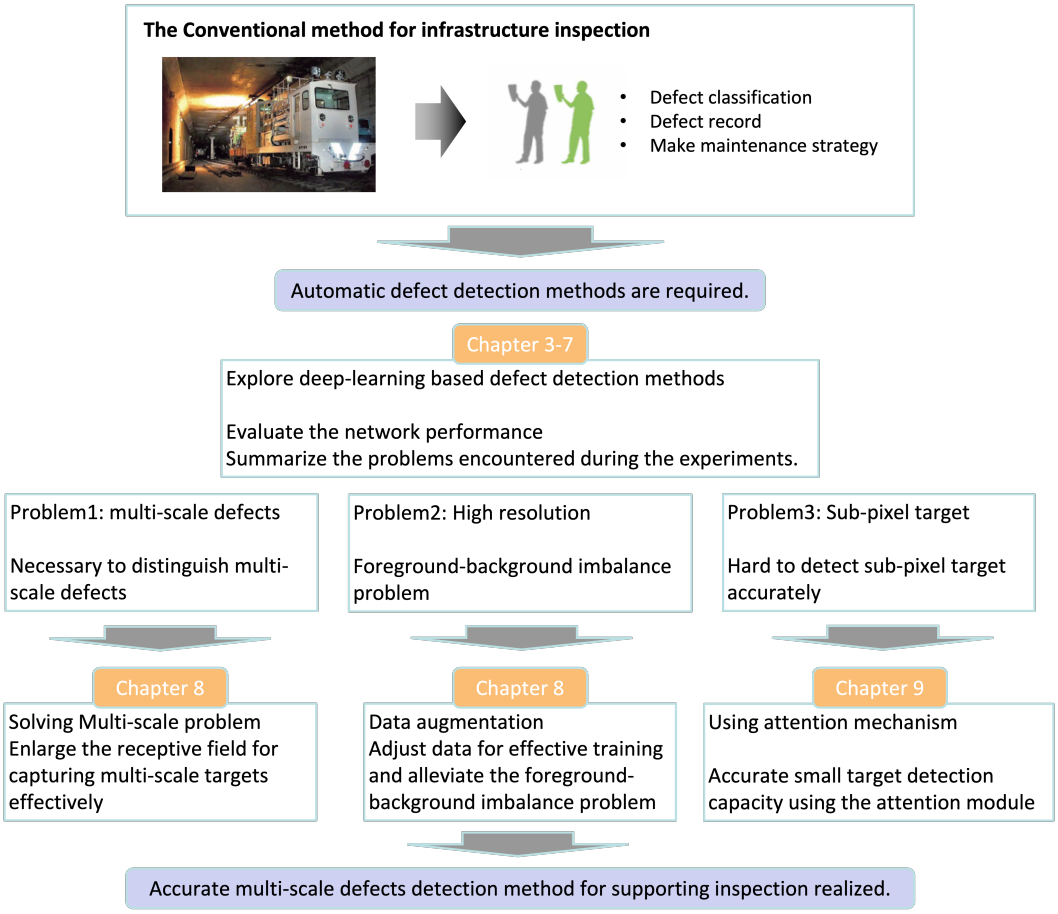


Figure 1.1: Overview of issues and solutions in this research.

Chapter 2

Related Works

2.1 Introduction

Firstly, the development of semantic segmentation methods is described in Section 2.2. Thereafter, I summarize previous studies on maintenance and management support for infrastructure facilities in Section 2.3 and summarize them in Section 2.4.

2.2 Review of semantic segmentation

Reference [13]

The authors addressed the limitations of traditional convolutional neural networks (CNN), which are primarily designed for image classification, by proposing an architecture that can perform pixel-level predictions. The key idea of FCN is to replace the fully connected layers of CNN with fully convolutional layers, thereby enabling the network to accept input images of arbitrary sizes and produce dense predictions with the same spatial dimensions as the input. This is achieved by adjusting the stride of the convolutional layers to one and employing appropriate up-sampling techniques to restore the output resolution.

Reference [14]

SegNet architecture is built on a CNN with an encoder-decoder structure. The encoder network consists of multiple convolutional and pooling layers that progressively extract and encode hierarchical features from the input image. These layers help to capture both low-level details and high-level semantic information. The key advantage of SegNet is its

high memory efficiency. By using pooling indices for up-sampling, SegNet avoids storing redundant information and significantly reduces memory requirements compared with other segmentation networks.

Reference [15]

U-Net architecture is structured as an encoder-decoder network with skip connections. The encoder module captures high-level semantic features through a series of convolutional and pooling layers, progressively reducing the spatial resolution while increasing the depth of feature maps. U-Net demonstrated exceptional performance in numerous segmentation tasks, particularly in scenarios with limited training data. Its ability to capture both local details and global context has made it especially effective in biomedical image segmentation, where the precise delineation of structures is critical. U-Net applications include cell segmentation, tumor detection, and organ segmentation.

Reference [16]

RefineNet is a multi-path refinement network designed for high-resolution semantic segmentation. By incorporating coarse-level and fine-level sub-networks and iteratively refining predictions, RefineNet effectively captures global and local information, resulting in accurate and detailed segmentation results. Specifically, at each stage, the coarse-level sub-network takes as input low-resolution feature maps and produces initial predictions. These predictions are then up-sampled and combined with the fine-level sub-network, which operates on higher-resolution feature maps. The fine-level sub-network further refines predictions by capturing more detailed information and incorporating fine-grained spatial context.

Reference [17]

PSPNet is a pyramid scene parsing network that captures multi-scale contextual information using pyramid pooling modules. It achieves accurate segmentation by integrating global and local contexts. The backbone network extracts hierarchical features by combining low-level details with high-level semantics. The pyramid-pooling module aggregates information from different scales by dividing the feature maps into regions and applying pooling with various window sizes. The ability of PSPNet to incorporate both local and global contexts contributes to its influence on semantic segmentation.

Reference [18]

DeepLab v2 is a deep CNN designed for semantic image segmentation. It employs atrous convolution, dilated convolutions, and fully connected CRFs to capture context and refine segmentation results. With an encoder-decoder architecture, it extracts high-level semantic features using atrous convolution in the encoder and performs up-sampling in the decoder. The network incorporates dilated convolutions and post-processing with CRFs for improved accuracy along object boundaries and efficient handling of large receptive fields, which make DeepLab v2 a classic and influential network in the field of semantic segmentation.

Reference [19]

DeepLab v3+ is a renowned semantic segmentation network that improves accuracy and efficiency. It uses an encoder-decoder architecture with atrous separable convolutions. The encoder captures high-level features, while the feature pyramid network enhances representation. Atrous separable convolutions reduce parameters. The decoder performs up-sampling and incorporates skip connections.

Reference [20]

DANet is a classic semantic segmentation network known for its dual attention mechanism, which captures both global and local dependencies. It incorporates spatial and channel attention modules to enhance feature representations and improve segmentation accuracy. DANet has achieved great performance in semantic segmentation tasks, making it a significant network architecture in the field.

Reference [21]

HRNet is a classic semantic segmentation network known for its ability to handle high-resolution input images effectively. It maintains high-resolution representations throughout the network using parallel sub-networks and a multi-resolution fusion strategy. By preserving fine details and combining local and global context, HRNet achieves state-of-the-art performance in semantic segmentation tasks.

Reference [22]

SegFormer is a notable semantic segmentation network that combines transformers with CNN. It captures global context and dependencies using self-attention mechanisms while

maintaining computational efficiency. By fusing local and global information and leveraging a hierarchical segmentation head, SegFormer has demonstrated competitive performance in this field.

Reference [23]

SwinTransformer is a cutting-edge semantic segmentation network that incorporates transformers into computer vision tasks. It utilizes a hierarchical structure and shifted windows to efficiently capture local and global dependencies. SwinTransformer achieves accurate segmentation by modeling fine-grained details and high-level context.

Reference [24]

Mask2former is a classic semantic segmentation network that combines mask-based methods with transformers. It introduces a layered self-attention mechanism to capture local and global context, improving segmentation accuracy. By incorporating iterative mask propagation and transformer encoders, Mask2former achieves state-of-the-art performance in dense image prediction tasks.

Reference [25]

Segment everything model (SAM) is a novel instance segmentation model which gains the capacity to segment any object in any image. SAM can be used as a foundation model for image segmentation, as it can be easily adapted to different domains and tasks with simple prompts. SAM is also efficient and scalable, as it can run on a single GPU and handle high-resolution images.

2.3 Review of deep-learning method based on real-world dataset

Reference [26]

An automatic defect detection and classification method using tunnel images taken from a charge-coupled device camera is proposed. Specifically, the acquired images are input to the FCN model to compute image features, and then the obtained image features are used in the RPN (a network that estimates candidate object regions) [27] and Position-sensitive RoI pooling [28], which is a network that estimates candidate object regions (RPN), and

performs defect detection. The authors show that the method has higher detection efficiency than Fast R-CNN [27], and can detect and classify multiple types of tunnel defects.

Reference [29]

The authors constructed a method for detecting defects in visualized images taken from subway tunnels. In this method, the obtained images are first divided into patches, and then the divided patches are input to CNN (LeNet) [30] for defect detection. Experimental results show that the method achieves higher accuracy in defect detection than the method using image features for identification.

Reference [31]

The authors constructed a CNN-based method for detecting cracks from visualized images. In this method, the authors constructed a novel detector based on CNN (MatConvNet) [32] to detect cracks with higher accuracy, because the detection problem is improved by the external environment such as light exposure in conventional edge detection studies.

Reference [33]

The authors proposed a method for detecting defects in subway tunnels using laser data. In this method, defect detection is performed using a full convolution network in 2D unfolding images of tunnel walls acquired by a 3D laser. Subsequently, the authors employed a loss function specifically designed to address class imbalance during the network training process. This approach effectively minimized the learning focus on image regions devoid of defects, which are abundant, thereby achieving exceptional accuracy in pixel-level defect detection.

Reference [34]

The authors proposed a crack detection method using deep learning in images captured from a high-resolution camera of an unmanned aerial vehicle (UAV). Specifically, the UAV is used to generate a model based on point clouds, and then the generated model is input to the R-CNN [35] of transition learning to detect and survey cracks on the surface of structures. Field tests show the effectiveness of the method.

Reference [36]

A fast defect detection and analysis system in subway shield tunnels is proposed in this paper. Specifically, the system detects and quantitatively analyzes defects in highly accurate tunnel surface images captured from a multi-array CCD camera using intelligence analysis technology. Then, based on the automatic detection results, the type, morphology, and distribution characteristics of the structural damage are analyzed, and the causes and factors affecting the structural damage are concluded.

Reference [37]

The authors proposed a system to inspect and measure cracks in concrete structures and to provide objective crack data for safety assessment. Specifically, the system consists of a robot system and a crack detection system. The robot system is controlled to maintain a certain distance from the wall while acquiring image data with a charge coupled device (CCD) camera. The crack detection system used image processing to extract crack information from the acquired images. To ensure accurate crack recognition, the geometric properties and patterns of cracks in the structure were applied to the image processing routines. The proposed system has been validated in the laboratory and in actual tunnel experiments.

Reference [38]

A defect detection method using texture analysis was proposed. In this method, various features are calculated from the concentration co-occurrence matrix calculated for each pixel. The method calculates various features from the density co-occurrence matrix calculated for each pixel, and classifies each pixel using a nonlinear Support Vector Machine [39] to estimate the area of defects. Visualization of subsurface rock structure using drilling data.

Reference [40]

The authors proposed a method for estimating the defect area from images of defects (mainly cracks) on roads. The proposed method uses three types of texture features: features obtained by morphological transformation, features obtained by Fourier transform, and features obtained by applying Steerable filter [41], respectively. AdaBoost is used to learn the obtained features to enable highly accurate region estimation.

Reference [42]

The authors proposed a method for estimating the defect area from images of structural

defects. By using color features, texture features, geographic information (e.g., symmetry and shadows of the captured area), and spatial information (e.g., the existence of a center line in the middle of a paved road) as features to be used, the method is able to estimate the defect areas of various types of structures. The system is able to estimate various defect areas of various structures.

Reference [43]

An edge detection method for road structure defects was proposed. The atrous algorithm enables the accurate removal of artificial noise in the defect image.

Reference [44]

The identification of fissures in asphalt surfaces via advanced image processing techniques was proposed. In this method, contrast enhancement is achieved through gray-scale transformation, median filtering, and histogram homogenization. These steps facilitate efficient noise reduction and edge detection. Additionally, accurate crack detection is achieved by employing segmentation through binarization.

Reference [45]

A structural inspection method employing a convolutional neural network based on fast regions (Faster R-CNN [27]) was proposed for detecting multiple types of defects. It is also compared to methods based on the conventional CNN. Considering that the proposed method provides a very fast testing speed (0.03 seconds per image at a resolution of 500 $\times$ 375 pixels), a framework for quasi-real-time damage detection in video using trained networks has been developed.

Reference [46]

A crack detection method in steel bridges using an infrared camera was proposed for highly accurate nondestructive evaluation of structures. The effectiveness of this method is demonstrated by nondestructive evaluation of structures using infrared thermography.

Reference [47]

The authors proposed an automatic image processing method for detecting cracks in concrete structures. The method includes two steps: (1) development of image filters to detect

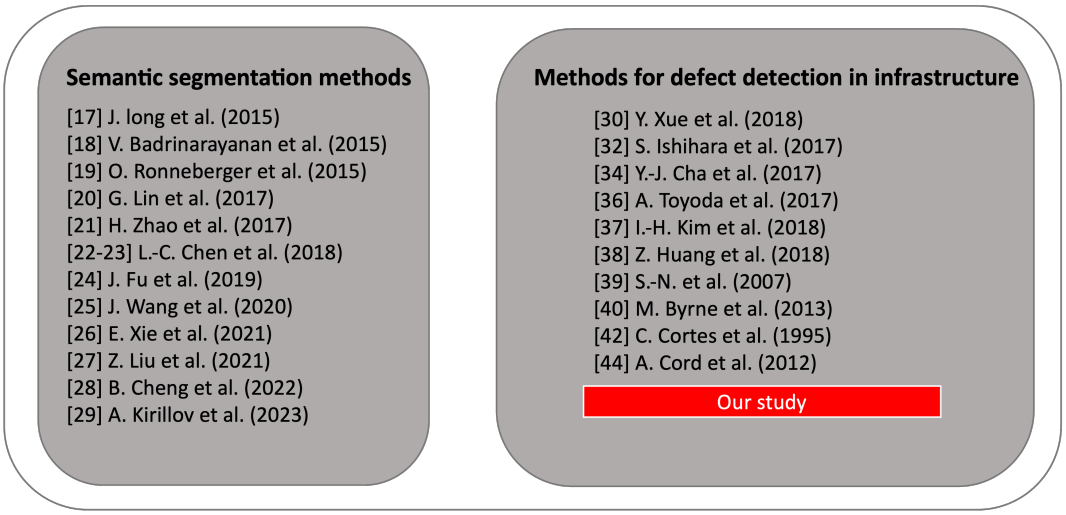


Figure 2.1: Research map of related studies.

major cracks using genetic programming, and (2) filtering out obscure cracks and removing residual noise after detection by iteratively applying the image filter to local regions around the cracks. As a result of the above, the proposed method can accurately detect cracks in structures recorded under various conditions.

2.4 Summary

Based on related research, the primary aim of this thesis is to utilize the semantic segmentation method to address the challenges and gaps in earlier inspection-supporting technologies. To achieve this goal, I made several improvements and conducted experiments in data augmentation and network structure design. The effectiveness of the proposed method will be evaluated using high-resolution images of subway tunnels. This thesis aims to demonstrate the performance improvements of the proposed method in real-world applications through experiments. Figure 2.1 shows a research map that summarizes the related research described above.

Chapter 3

Dataset in Research

3.1 Introduction

This chapter describes the inspection data used in this thesis. During the actual inspection, the inspector saves a visual image from the defect site as the inspection data. In the following section, the characteristics of the defects are explained.

3.2 Defect evaluation

This section describes the defect assessment during the inspections. During routine inspections of road structures, the condition of the structure is evaluated through close visual inspection. The engineers must record both their own name and the name of the person inspecting the road structure, and take close-up and far-away images of the location where the change has occurred. Finally, the captured images of the altered state and inspection records are used to create an inspection report. Subsequently, the inspection results are discussed based on the images and records. Following these evaluations, appropriate maintenance and management measures are taken.

For the dataset used in this study, high-resolution subway tunnel images captured by a special vehicle equipped with multiple sensors and cameras are used, and each image is stitched using different tunnel surfaces. Different representatives of these images come from different subway lines, and their construction periods and maintenance statuses are different; therefore, the difficulty of spot inspection during the maintenance process is also different.

In most concrete structures, the main defects in subway tunnels are categorized into three types: cracks, peeling, and water leakage. In tunnels, peeling is considered to be a major hidden danger

that threatens the safe operation of railways and is the key object of investigation, while other defects such as cracks and water leakage are rated according to the following criteria:

A1: major hidden dangers: measures must be taken immediately to repair.

A2: potential safety hazard, and repair as soon as possible.

A3: there is a tendency to become a threat and take repair measures when necessary.

B: leaving it alone may develop into the A level and focus on monitoring when necessary.

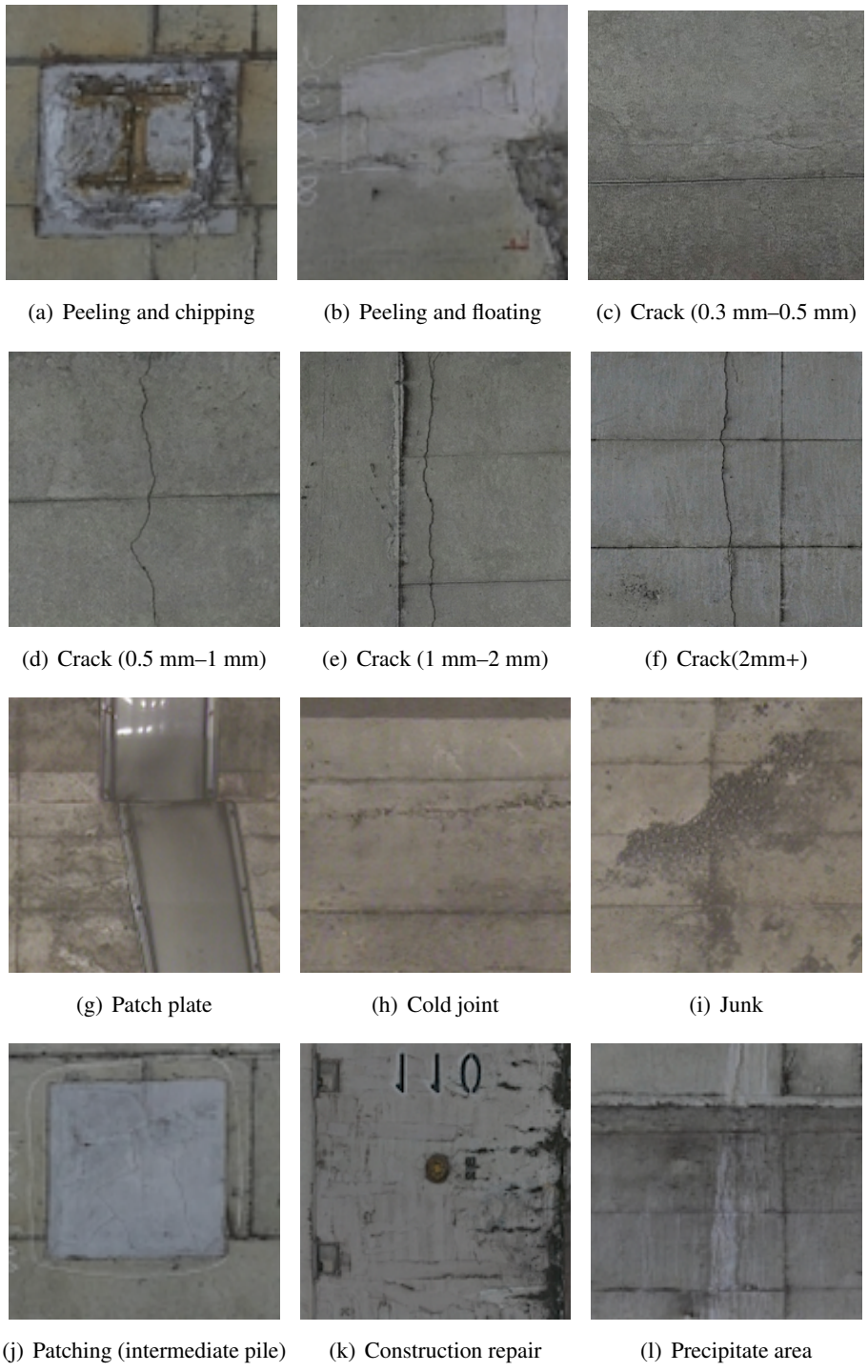
C: little impact on the status.

D: no defects.

Table 3.1 lists the defects that exist in the subway tunnel, whereas Fig. 3.1 shows the corresponding anomalous damaged areas and their structure types. Figs. 3.2- 3.5 show other examples.

Table 3.1: Defects exits in subway tunnel.

Number	name
01	Peeling and chipping
02	Peeling and floating
03	Crack (0.3 mm–0.5 mm)
04	Crack (0.5 mm–1 mm)
05	Crack (1 mm–2 mm)
06	Crack(2mm+)
07	Patch plate
08	Cold joint
09	Junk
10	Patching (intermediate pile)
11	Alligator crack
12	Repair of Deterioration
13	Decorative panel
14	Construction repair
15	Precipitate area
16	Masonry joint
17	Exposed reinforcing steel
18	Water leakage
19	Construction repair



(a) Peeling and chipping (b) Peeling and floating (c) Crack (0.3 mm–0.5 mm)

(d) Crack (0.5 mm–1 mm) (e) Crack (1 mm–2 mm) (f) Crack(2mm+)

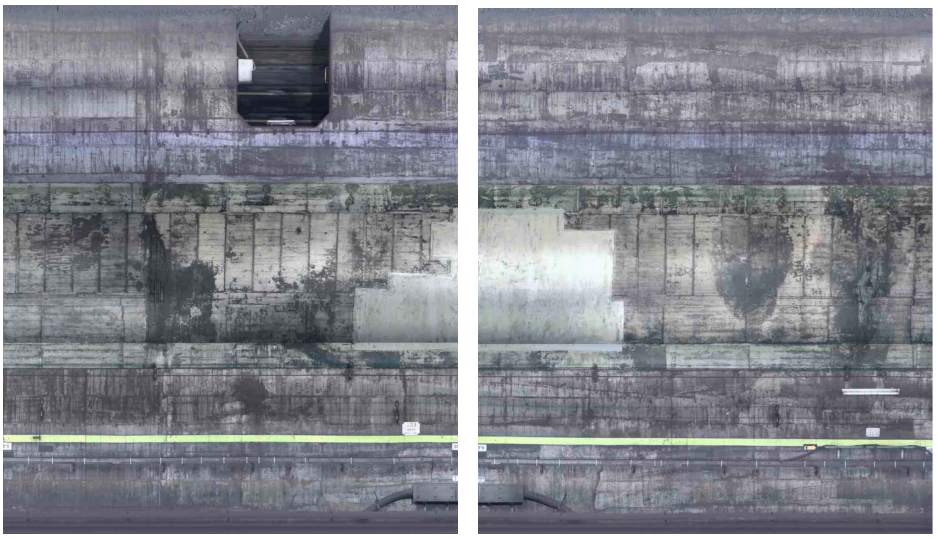
(g) Patch plate (h) Cold joint (i) Junk

(j) Patching (intermediate pile) (k) Construction repair (l) Precipitate area

Figure 3.1: Examples of defects existing in tunnel shield structure.

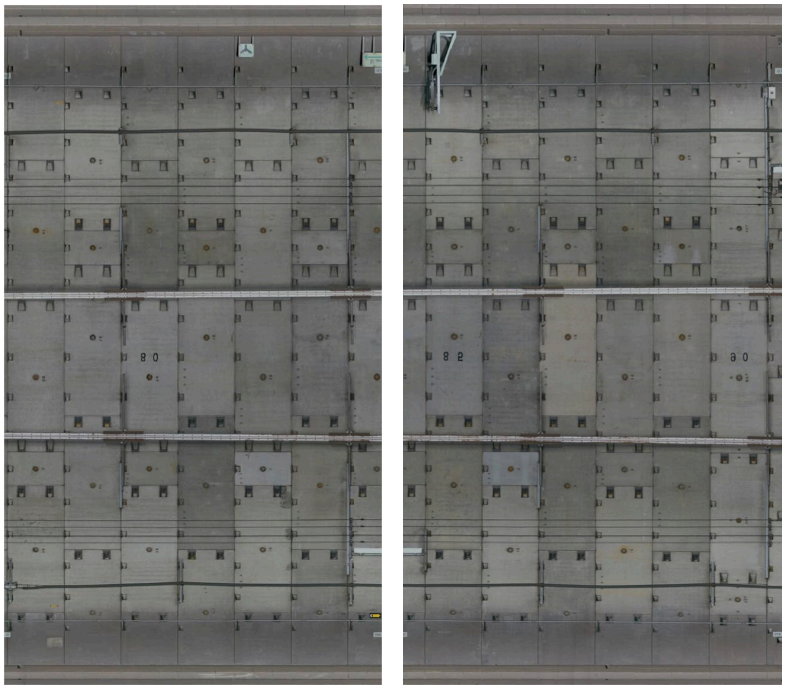
(a) (b)

Figure 3.2: Example of image version 1.



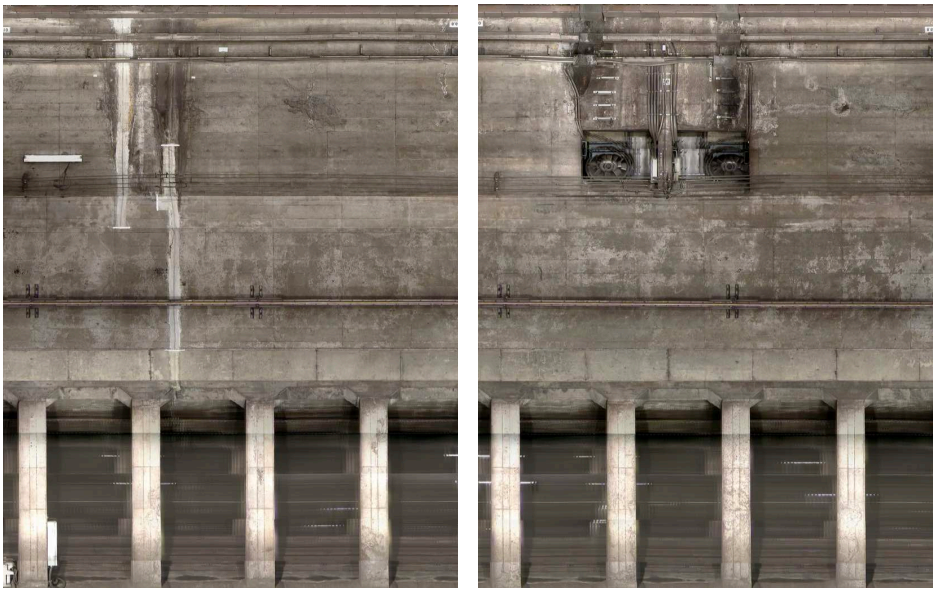
(a) (b)

Figure 3.3: Example of image version 2.



(a) (b)

Figure 3.4: Example of image version 3.



(a) (b)

Figure 3.5: Example of image version 4.

Chapter 4

Defect Detection Method in Subway Tunnels Based on CNN Method

4.1 Introduction

In this chapter, a method for classifying defects using the CNN approach is proposed. Although many methods utilize image features for defect detection in previous detection techniques, they generally detect only a single category of defects. To achieve high-precision detection of multi categories defects, I employ deep learning to construct a defect classifier. In Section 4.2, an overview of the CNN and residual module is introduced. The training and defect detection methods are detailed in Sections 4.3 and 4.4 respectively. The effectiveness of the proposed method is evaluated experimentally in Section 4.5. Finally, a summary is provided in Section 4.6.

4.2 Overview of CNN and residual module

CNN is a forward-propagating neural network. Its typical structure interweaves the convolutional and pooling layers, and eventually connects them to fully connected layers. The main process involves extracting features from input pixels through a hierarchical structure composed of convolutional and pooling layers before classifying them with fully connected layers.

Specifically, each layer performs the following calculations. In the convolution layer, the input $X^l \in R^{W\times W\times K}$ of size $W \times W$ with K channels obtained from the immediately preceding layer l is combined with M rectangular filters F^l of size $H \times H$. And each element of the input X^l is $x^l_{i,j,k}(i = 1, 2, \cdots, W, j = 1, 2, \cdots, W, k = 1, 2, \cdots, K)$ and each element of the rectangular filter is

$h^{(l)}_{p,q,m}(p = 1, 2, \cdots, H, q = 1, 2, \cdots, H, m = 1, 2, \cdots, M)$. Each element $x^{(l+1)}_{i,j,m}$ of the output from the m th filter to the $l + 1$ th layer is calculated using the following equation:

$$x^{(l+1)}_{i,j,m} = f(z^{(l+1)}_{i,j,m}), \tag{4.1}$$

$$z^{(l+1)}_{i,j,m} = \sum_{k=1}^{K} \left\{ \sum_{p=1}^{H} \sum_{q=1}^{H} x_{s\cdot i+p, s\cdot j+q, k} h^{(l)}_{p,q,m} \right\} + b_m, \tag{4.2}$$

where $f(\cdot)$ denotes the activation function. In addition, b_m represents the bias, which is often dependent only on the filter m. In addition, s represents the filter interval movement (stride), where the stride is large. Consequently, the output size is decreased.

Next, in the pooling layer, the output of the convolutional layer is used as the input, producing one output value from its local region. Specifically, when the input $X^{(l)} \in R^{W\times W\times K}$ of size $W \times W$ with k channels is derived from the previous convolutional layer in the l th layer, a $P \times P$ square area $P_{i,j,k} \in R^{P\times P}$ centered on the input element (i, j) in channel K is considered. From the values of j,k and, the output value $x^{(l+1)}_{i,j,k}$ is calculated using the following equation:

$$x^{(l+1)}_{i,j,k} = \max_{x^{(l)}_{i,j,k} \in P_{i,j,k}} x^{(l)}_{i,j,k}, \tag{4.3}$$

Eq. (4.3) describes max pooling, and the stride of $P_{i,j,k}$ is generally set to a value of 2 or more.

Finally, in the fully connected layer, when all the input values from the previous l th layer are $x^{(l)}_t(t = 1, 2, \cdots, T)$, the output value x^{l+1}_u for unit u in the $(l + 1)$ th layer is calculated using the following equation:

$$x^{(l+1)}_u = f(u^{l+1}_u), \tag{4.4}$$

$$u_u^{l+1} = \sum_{t=1}^{T} \left\{ \omega_{t,u} x_t^{(l)} \right\} + b_u, \tag{4.5}$$

where $\omega_{t,u}$ is the weight of the input value t when calculating the output value to unit u, and b_u is the bias. When the objective is classification, the number of units in the last layer is set as C, which is equal to the number of classes, and the input of the last layer $x_c(c = 1, 2, \cdots, C)$ is used to calculate the belonging probability p_c of each class and then calculate the classification results $\hat{y}$. The specific equations are as follows:

$$p_c = P[y = c|x_c] = \frac{exp(x_c)}{\sum_{c=1}^{C} exp(x_c)}, \tag{4.6}$$

$$\hat{y} = \arg\max_c p_c, \tag{4.7}$$

where y represents the true class label. As previously mentioned, the CNN extracts and classifies features from the input pixel values. This enables simultaneous calculation and classification of features that align with the objective.

Then, the characteristics of the residual network are introduced. The main contribution of the residual network is that it addresses the vanishing gradient problem during deep network training. Specifically, it uses two strategies, identity mapping and residual mapping, to address the degradation issue. Specifically, the deep network is structured as $H(x) = F(x) + x$, and the learning objective is to learn the residual function $F(x) = H(x) - x$. If $F(x) = 0$, this results in an identity map $H(x) = x$, making it simpler to fit the residuals. In other words, the network will continue to deepen, and $F(x)$ will be pushed to 0, leaving only identity mapping x. With this design, the network will always be maintained in the optimal state in theory, and it will not cause performance drops by increasing the network depth.

Table 4.1: The network of architecture with residual modules used in the proposed method.

Layer name	Output size	Layers
Conv 0	200×200	7×7, 32, stride 2
-	100×100	3×3, 32, stride 2
Conv 1	100×100	$\begin{pmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{pmatrix} \times 5$
-	50×50	3×3, 64, stride 2
Conv 2	100×100	$\begin{pmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{pmatrix} \times 5$
-	50×50	3×3, 128, stride 2
Conv 3	25×25	$\begin{pmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{pmatrix} \times 5$
-	1×1	average pool, 2-d fc, softmax

4.3 Training of defect detector using residual module based CNN network

First, patches of $v \times v$ pixels are calculated from the subway tunnel image, and a correct label indicating whether or not a defect is included is assigned to each patch. However, if a patch with a slight defect at the edge of the patch is given the correct label of "defect area," it will be difficult to distinguish it from a patch without any defect. Therefore, this situation should be avoided. I apply the folding processing of $\frac{v-d}{2}(d \leq v)$ pixels and determine whether the region of the center $d \times d$ pixels of the obtained patch contains a defect, through which the correct label can be determined. In addition, by setting the patch slide width such that the central areas of adjacent patches overlap, all defects are included in the central area of one of the patches. Next, I train the network, whose architecture is shown in Table 4.1, using defect patches to build a defect detector.

4.4 Defect detection in subway tunnel images

The subway tunnel images are divided into N patches in the same manner as the dividing method described above, and it is unknown whether defects are included. Furthermore, all the obtained patches are the previously learned residual modules. By defining $n(= 1, 2, \cdots, N)$ as the input, the probability p_n^+ that the patch contains defects and the defect detection result $\hat{y}_n$ are calculated.

Then, the calculated p_n^+ is assigned to the central region of $d \times d$ pixels of each patch. Depending on the slide width for calculating the patches, there is a case where the central areas of the patches overlap; therefore, the average of those probability values is given to the overlapped areas. Through the above processing, the proposed method enables the calculation of the probability value that each patch contains a defect and realizes the automatic detection of the presence or absence of defects in subway tunnel images.

4.5 Experiment

In this section, the proposed method is applied to subway tunnel images, and its detection accuracy is compared with that of conventional methods using FCN and CNN. In this experiment, 154,372 patches calculated from 10 images of subway tunnels are used as training data, and 102,960 patches calculated from 6 images of subway tunnels are used as test data. The size of each patch is 400×400 pixels ($w = 200$), and the size of the central area of the patch is 220 pixels (d=220). The sliding width of the patches is set to 100 pixels. The ResNet in this experiment is ResNet50, and the number of training epochs is 50.

The comparative methoda are the traditional FCN and CNN methods. Specifically, I construct an FCN that identifies whether wide-area defects, such as peeling or water leakage, are included. Next, a CNN is constructed to identify narrow-area defects, such as several types of cracks. The final defect detection result is obtained by sequentially inputting the images to be identified in the trained FCN and CNN. The model for the FCN is FCN-8s based on VGG16 [48] and fine-tuned with the PASCAL VOC 2011 dataset. For the CNN, I used LeNet [30] and employed dropout in the fully connected layer, where the units to be connected are selected to prevent over-fitting during the training. The detection recall for each method is listed in Table 4.2.

4.6 Summary

In this chapter, a defect detection method for images of subway tunnels using a CNN is proposed. The experimental results demonstrate that the method based on the ResNet architecture achieves higher detection accuracy than the traditional CNN and FCN models. However, the effectiveness of CNN and FCN needs to be further explored. In the next chapter, I aim to construct a

Table 4.2: Recall of each defect.

	ResNet	CNN(LeNet)	FCN
Peeling and chipping	**0.912**	-	0.236
Peeling and floating	**0.832**	-	0.224
Crack (0.3 mm–0.5 mm)	**0.794**	0.643	-
Crack (0.5 mm–1.0 mm)	**0.937**	0.900	-
Crack (1.0 mm–2.0 mm)	**0.956**	0.943	-
Crack (2,0 mm+)	0.908	**0.921**	-
Junk	**0.740**	-	0.312
Repair of deterioration	**0.933**	-	0.457
Construction repair	**0.913**	-	0.413
Masonry joint	**0.904**	-	0.276
Exposed reinforcing steel	**0.865**	-	0.144
Water leakage	**0.902**	-	0.336

composite model to verify the detection effectiveness of multi-scale targets of the FCN and CNN models.

Chapter 5

Defect Detection Method Based on CNN and FCN Method

5.1 Introduction

This chapter describes a method for detecting defects in subway tunnels using FCN and CNN. The previous chapter shows that the accuracy of the FCN model is insufficient, which does not match the characteristics of the FCN model. Therefore, in this chapter, I adjust the dataset and training method for the FCN model and further propose the defect detection method using the combination of CNN and FCN (hereinafter called "proposed method"). In the literature [29], the author demonstrated that training the CNN model with small patches can improve the detection capacity of small defect objects such as cracks. Still, it is hard to accurately detect large defect objects such as water leakage. Smaller patches mean larger training data and slower training speed. Considering computational efficiency and overall accuracy, the CNN model is tasked with detecting small defect objects and the FCN model is dedicated to identifying large defect objects. Specifically, I divide the subway tunnel image into large patches and construct an FCN model that discriminates whether large defect objects are included. The input image is then divided into small patches, and a CNN model is constructed to identify whether small defect objects are included. And in the test phase, the defect detection result is obtained by sequentially inputting the images to be analyzed to the trained FCN and CNN.

The defect detection method is explained in Section 5.2. And the Section 5.3 explains the experiment settings and results. Finally, Section 5.4 summarized this chapter.

5.2 Defect detection method in subway tunnels using FCN and CNN

The proposed method consists of the process of learning an FCN to detect large defect objects that appear in a wide area, learning a CNN to detect small defect objects, and detecting defects using the FCN and CNN. The following section describes the training of the FCN model for large defect object detection in Subsection 5.2.1, and the training of the CNN model for small defect object detection in Subsection 5.2.2. Finally, the testing phase explained in Subsection 5.2.3.

5.2.1 Learning FCN to detect large defect objects

The FCN replaces all fully connected layers of existing CNN model, such as AlexNet [49] with convolutional layers and converts all processing to filtering. Specifically, all fully connected layers that are directly connected to the convolutional or pooling layer are replaced by convolutional layers with a filter of the same size as the input and 1×1 kernel size. These replacements make constructing a network independent of input image size possible and allow the entire image to be input, thus enabling faster classification than classifying multiple patches using CNN. In addition, the FCN integrates deep and shallow layer outputs into the network to achieve more accurate pixel classification. Specifically, the outputs of the middle layer are repeatedly up-sampled, performed using bilinear interpolation to the same size and added together, and finally up-sampled to the same size as the input image for pixel-level classification. This enables a detailed understanding of the content and location of defects in the image. In addition, when training the above network, weights learned in existing networks for classification purposes can be used as weights, except for the replaced all-combining and added layers. Therefore, it enables efficient learning using the fine-tuning model.

In the following experiments, I define the following types of defects as those that appear over a wide area: peeling and chipping/floating, junk, patch plate, cold joint, patching (intermediate pile), repair of deterioration, precipitate area, construction repair, water leakage, and repair due to deterioration. The obtained image is then divided into multiple large patches $X_i \in R^{H \times W} (i = 1, 2, ..., I)$; I is the total patch number. By inputting each patch X_i from the subway tunnel image into the learned FCN, the estimation probability is $Y_i \in R^{H \times W \times (K+1)}$, where each pixel contains irregularities. The value of K is set to 1 for the discrimination of whether there is a defect. By

applying the above process to all patches, the identification result $Y_i \in R^{H\times W\times(1+1)}$ for the entire image of the subway tunnel is obtained. This process enables the classification of defects that appear over a wide area.

5.2.2 Learning CNN to detect small defect objects

CNN is a forward propagating conjunctive neural network consisting of alternating convolutional and pooling layers, followed by multiple fully connected layers. By inputting small size patches into the CNN, it is expected to be able to detect small defect objects. In the experiments that follow, I define the following defects as those that appear in narrow areas: delamination/flaking, cracks, cold joints, specific-shaped cracks, decorative panels, openings, and joints. The proposed method divides the subway tunnel image into small patches $\hat{X}_i \in R^{\hat{H}\times\hat{W}}(i = 1, 2, \ldots, i; i$ present the number of total patches). However, $\hat{H} < H$ and $\hat{W} < W$ to make the input patch size smaller than the FCN input patch.

In the proposed method, I construct a CNN model to detect defects for each patch, similar to that in the literature [29]. In this case, the sliding width of the patches is set such that the central regions of adjacent patches overlap so that all defects are included in the central region of one of the patches. However, if the central regions of the patches overlap during the test, the average probability of being an irregular region is assigned to the overlapped region. This method above enables the detection of small detect objects.

5.2.3 Defect detector of proposed method

The proposed method inputs test images to the FCN and CNN to obtain the final defect detection results. Specifically, patch segmentation is first applied as in Subsection 5.2.1, and then each patch is input to the FCN to detect defects that appear over a wide area. Then, patch segmentation strategy in 5.2.2 is applied, and patches with less than $T\%$ of the area estimated to be defective by CNN are selected. It should be noted that $T = 30$ is set in the subsequent experiments. The defects that appear in the narrow region are detected by inputting the selected patches into the CNN. The method outlined allows us to detect both large and small defect objects in the test image.

Table 5.1: The detection results of each model.

	Mean acc	Mean IoU	Recall	Precision	F-value
FCN	0.493	0.455	-	-	-
CNN	-	-	0.463	0.331	0.386

5.3 Experiment

In the experiment, 596 tunnel images are used as training data and 48 as test data. Specifically, for the FCN, 321,435 patches are used as training data, and 138,397 patches are used as test data. The size of each patch is set to 500×500 pixels, and the sliding width of the patches is 250 pixels. The FCN model is FCN-8s based on VGG16 [48], and the PASCAL VOC2011 dataset is used for fine-tuning the pre-trained model. For the CNN, a total of 7, 994, 732 patches are used as training data and 3,542,375 patches as test data. The T mentioned in the preamble is set to 30. The size of each patch is set to 200×200 pixels, and the size of the central area of the patch is set to 110 pixels. The sliding width of each patch is set to 50 pixels. ReLU is used as the activation function of the CNN, and a dropout method is employed in the fully connected layer to select the units to be combined, preventing over-fitting.

In the experiments, different metrics are applied to evaluate the detection capacity of both models. Because the FCN model performs defect detection on image pixels, the detection accuracy of the proposed method is evaluated using the mean acc and mean IoU according to the literature [13]. For the CNN model, which performs defect detection capacity on the patches, the detection accuracy of the proposed method is evaluated using precision, recall, and F-measure, as described in the literature [8].

The experimental results are shown in Table 5.1. The experiment results show that the FCN and CNN models perform well with defect detection capabilities in their respective tasks. Therefore, the effectiveness of the proposed method is demonstrated.

5.4 Summary

In this chapter, I propose a method for defect detection in subway tunnel images using FCN and CNN. Specifically, the proposed method uses an FCN model to classify large defect objects and a CNN model to classify small defect objects. Although the overall accuracy has been improved to

a certain extent, the training part of the CNN requires too much training data and a long training time, and using two models at the same time is not conducive to application. Therefore, in the next chapter, defect detection using only the FCN is explored.

Chapter 6

U-Net-based Segmentation Method for Defect Detection

6.1 Introduction

In this chapter, a defect detection method for subway tunnels based on U-Net is proposed. U-Net [15] is based on VGG16 [50], and it comprises a fully convolutional network that adopts a structural decoder. U-Net is capable of learning using an arbitrary image size as input, segmenting regions in units of pixels, and storing spatial position and detailed information by combining features. Since U-Net has such properties, it is considered effective for objects other than general images. Therefore, this chapter proposes a defect detection method for subway tunnel images based on U-Net with the aim of realizing pixel-level defect detection. Pixel-level defect detection is realized by dividing the image into small regions (patches) and constructing a network that identifies whether each patch contains defects, such as cracks and water leaks pixel-level defect detection is realized.

Section 6.2 explains the defect detection method using U-Net. Experiments and discussion are given in Section 6.3. The summary of this chapter is presented in Section 6.4.

6.2 Defect detection method using U-Net

In the proposed method, a U-Net-based method is constructed to realize pixel-level defect detection. U-Net is a fully convolutional network that adopts a structure called encoder-decoder. Specifically, U-Net consists of three networks: encoder, bridge, and decoder, which alternately

Table 6.1: The performance of the proposed method and the three comparative methods.

Model	IoU	Recall	Precision	F-value
Proposed Method	**0.325**	0.419	**0.592**	**0.491**
SegNet [14]	0.113	0.092	0.553	0.158
FCN full-training [13]	0.220	0.347	0.530	0.419
FCN fine-tuning [13]	0.239	**0.495**	0.315	0.385

connect convolutional and pooling layers. A bridge is constructed only from convolutional layers and realizes the connection between the encoders and decoders. The decoder is a network in which an up-sampling layer, a feature connection layer, and a convolution layer are alternately connected, and it is possible to store spatial position information using the feature connection layer.

In the proposed method, the following two changes are made to U-Net to realize defect detection from subway tunnel images.

Novel point 1: Add 1 convolutional layer to all encoder and decoder convolutional blocks. Also 3 convolutional layers and 1 max-pooling layer are added to the encoder and decoder input sides, respectively. Consequently, the depth of the network is improved, and it becomes possible to capture more detailed defect features.

Novel Point 2: Set the maximum value of the feature map channel to 256. Generally, in U-Net, during the upsampling process of the decoder, the output of each upsampling is combined with the low-dimensional features obtained from the encoder. By changing this size, the number of learning parameters can be reduced, which can increase the training speed.

Thus, it is possible to construct a U-Net model that can detect defects in both narrow and wide areas with high accuracy.

6.3 Experiment

In this experiment, 268,170 patches obtained by dividing 30 subway tunnel images (hereinafter referred to as images) are used as the training data. The size of each image is 10,000×12,088, 10,000×12,588, and 10,000×13,488 pixels. In addition, 71,818 patches obtained from 6 images are used as validation data, and 356,048 patches obtained from 12 images are used as test data. The size of each patch is 256×256 pixels and the slide width is 64 pixels.

I compare the accuracy of the proposed method with the fully convolutional network adopted in multiple semantic segmentation tasks that have been proposed in recent years. The accuracy of full training and fine-tuning is also examined. In this experiment, 20 epochs of learning are performed for all methods. Recall rate, precision rate, F-value, and IoU are used for accuracy evaluation.

Table 6.1 lists the defect detection accuracy of each method. The experimental results demonstrate the effectiveness of the proposed method for detecting defects in subway tunnel images. As shown in Fig. 6.1, it is confirmed that the U-Net of the proposed method is superior to that of the comparison method in terms of the detection accuracy of narrow-area defect and wide-area defect. Based on the above results, the following conclusions are obtained.

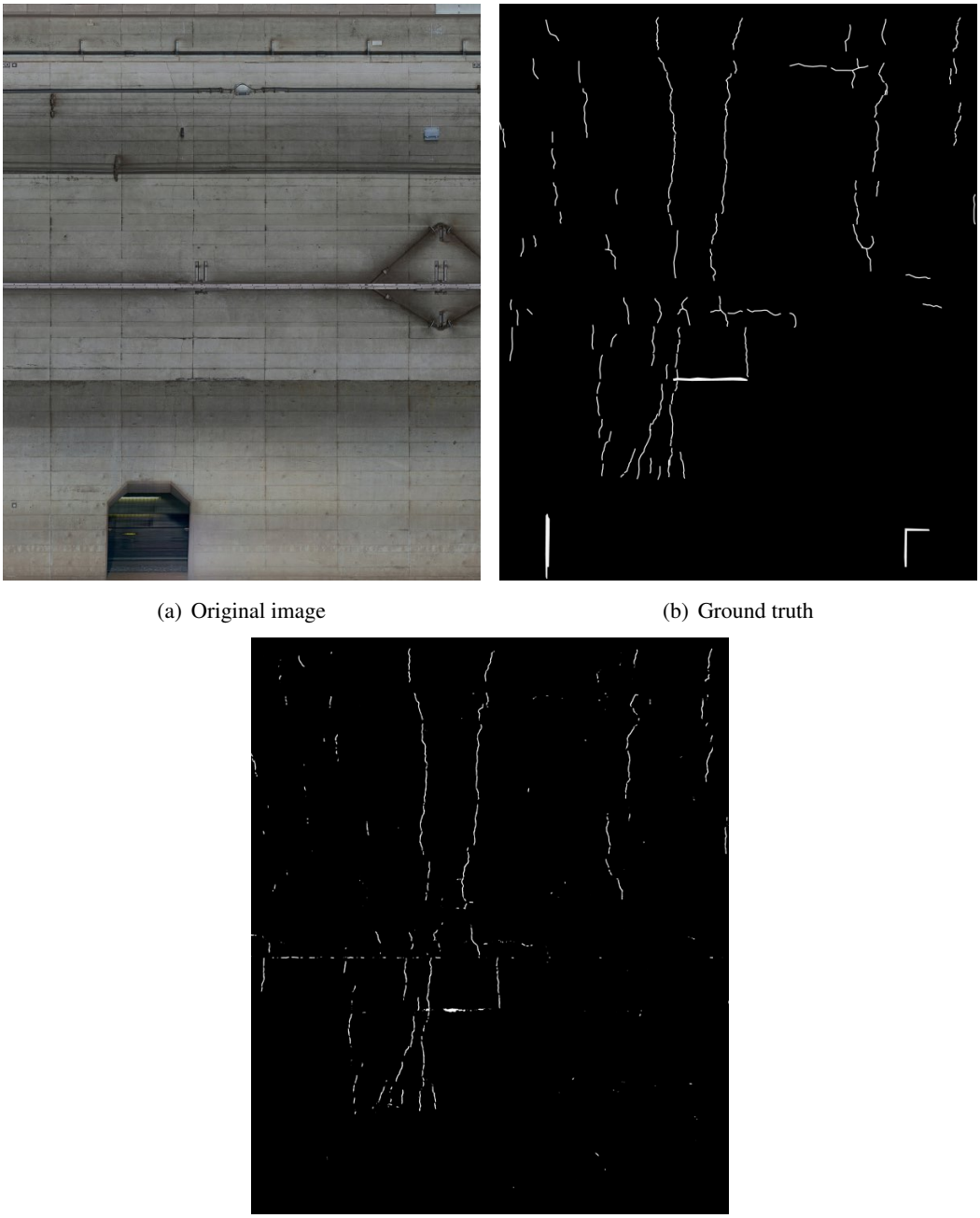
1. By adopting a fully symmetrical encoder-decoder structure of U-Net, I can obtain higher detection accuracy than FCN and DeepLab v3+ for multi-shape defects.

2. The effectiveness of using a feature combination layer is demonstrated because the proposed method had higher accuracy than SegNet, which employs an encoder-decoder.

3. A comparison between U-Net and ResU-Net confirmed the effectiveness of selecting VGG for the network structure.

6.4 Summary

In this chapter, a defect detection method for subway tunnel images based on U-Net is investigated. The experiments demonstrated the effectiveness of the proposed method. To explore the effectiveness of U-Net, in the next chapter, more comparative methods will be added, and the advantages and problems of these methods will be analyzed.



(a) Original image (b) Ground truth

(c) Estimated region

Figure 6.1: An example of detection result of my method. (a) the original image, (b) the ground truth region image and (c) the estimated region image.

Chapter 7

Further Validation of U-Net-based Defect Detection Method

7.1 Introduction

In this chapter, defect detection in subway tunnels based on semantic segmentation is studied. The proposed method is explained in Section 7.2. Experiments are conducted to confirm the effectiveness of the proposed method in Section 7.3. The results and findings are summarized in Section 7.4.

7.2 Defect detection based on semantic segmentation method

In this section, I describe my proposed defect detection method based on semantic segmentation. Since subway tunnel images are high-resolution, the learning phase of the proposed method is based on patch segmentation, and the defect regions are learned for each patch. Specifically, patches are segmented to overlap each other by a certain width, and these segmented patches are input into a network using various semantic segmentation methods to learn the defects at the pixel level.

Next, in the test phase, I perform patch segmentation as in the previous method, and input all obtained patches into the trained network. This is followed by pixel-level defect detection for each patch. However, when merging the results obtained from each patch, the superimposed areas caused by the slide width are assigned the average of the probability values for the superimposed pixels. This process results in pixel-level defect detection.

In the subsequent sections, each network employed in the learning process is described.

7.2.1 Fully convolutional network (FCN)

In FCN [13], all processing is converted to filtering by replacing all coupling layers of existing CNNs such as AlexNet [12] and GoogLeNet [50] with convolution layers. Specifically, I replace all the coupling layers with convolutional layers that have a filter of the same size as the input. In addition, the convolutional layer with a filter size of 1×1 is used between all the coupled layers. Note that the number of units of all joined layers is the number of filters in the convolutional layer. The above replacement enables us to construct a network independent of the size of the input image.

Furthermore, FCN introduces the integration process of the outputs of the deep layer and the shallow layer into the network in order to achieve more accurate pixel classification. Specifically, the outputs of the middle layer are repeatedly up-sampled and added, and finally up-sampled to the same size as the input image to perform pixel-level classification. The up-sampling is performed by bi-linear interpolation. This allows us to obtain detailed information about the defects and their locations in the image. In the training of the above network, the weights other than those of the replaced all-joining layers and the added layers can be the weights learned in the existing networks for classification purposes. Therefore, fine tuning enables efficient learning.

7.2.2 U-Net

U-Net [15] is an all-convolutional network that employs a structure called encoder-decoder. Specifically, U-Net consists of three networks called encoder, bridge, and decoder. In encoder, several convolutional layers and pooling layers are alternately connected. The bridge is constructed from only convolutional layers and realizes the connection between the output of encoder and the input of decoder. The decoder is a network of alternating connections among the up-sampling layer, the feature combination layer, and the convolutional layer, in which the convolutional layers of the encoder and the decoder are symmetric. U-Net is characterized by the preservation of spatial location information by feature combination. The feature maps output from each layer of encoder are directly concatenated with the feature maps of the corresponding layer of decoder to

capture the detailed information of pixel-level regions.

This section describes the difference between the network structure of U-Net and FCN. The U-Net and FCN approaches are similar in that they integrate feature maps from the previous layer. In FCN, the feature maps of different layers are merged by adding the values of each channel, whereas in U-Net, the feature maps output by encoder are added to the feature maps of decoder as a separate channel. At the same time, the network structure of FCN is different from that of U-Net in that FCN is not symmetrical, and there is a loss of detailed information. Based on these differences, FCN and U-Net are distinct.

Residual U-Net [51] is a model in which all convolutional units of the U-Net are modified with residual blocks, and the residual structure is effective in preventing gradient loss even when the layers are deep.

7.2.3 SegNet

SegNet [14] has an encoder-decoder structure similar to that of U-Net. However, unlike U-Net, the network structure uses a mean pooling layer instead of a feature combination layer.

7.2.4 DeepLab v3

DeepLab v3+ [19] extends DeepLab v3 by adding a simple and effective decoder module for recovering object boundaries. DeepLab v3 is an all-convolutional network built on ResNet [52]. Specifically, the ASPP structure is connected to the output of the convolutional layer of ResNet, and the encoder-decoder structure is applied. The ASPP structure is a dilated convolution model that extends the receptive fields in the feature map. This structure is considered to be especially effective in detecting wide-area defects when context information is captured at multiple scales.

7.3 Experiment

In this section, experiments are conducted to verify the effectiveness of each semantic segmentation method. 7.3.1 describes the experimental conditions and 7.3.2 presents the experimental results.

7.3.1 Experiment settings

In this experiment, I use 30 subway tunnel images (hereafter referred to as images) provided by Tokyo Metro Co., Ltd. A total of 268,170 patches obtained by dividing the total number of images into segments are used as training data. The training data consists of patches both with and without pixel-level variables. Each patch is annotated with the presence or absence of pixel-level defects. 71,818 patches from 6 images are used as validation data, and 356,048 patches from 12 images are used as test data. The resolution of each image is either 10,000 × 12,088, 10,000 × 12,588, or 10,000 × 13,488 pixels. For the patch segmentation, the size of each patch is 256 × 256 pixels, and the slide width is 64 pixels. In training, data expansion by Random Cropping and Random Flipping is performed. The size for cropping is set to 224 × 224 pixels. I used the stochastic gradient descent method as the learning algorithm for all-layer convolutional networks, and the value of the parameter of the Momentum term is set to 0.9. In order to suppress overlearning, WeightDecay is used, and its value is set to 0.001. And the learning rate is set to 0.001. In my experiments, the batch size is set to 16, and the number of learning epochs is set to 20.

I evaluated the detection accuracy by calculating Intersection over Union (IoU), Recall, Precision, and F-value as shown in the following equations, based on the estimation results for the test data.

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{7.1}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{7.2}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{7.3}$$

$$\mathrm{F-value} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}, \tag{7.4}$$

where TP, FP, and FN are the number of true positive, false positive, and false negative pixels, respectively, when the class with defect is taken as a positive example.

I apply several recently proposed semantic segmentation methods to this task and compare their accuracies. I also examine the accuracy of FCN, the most basic of the semantic segmentation

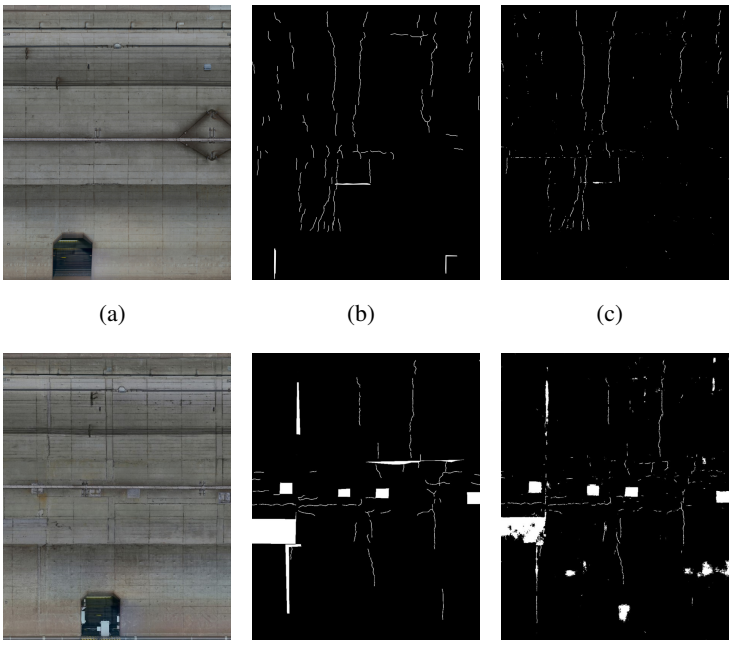


Figure 7.1: The detection result of ground truth and U-Net. (a), (d) present original image. (b), (e) present the ground truth, (c), (f) is the detection result of each images.

methods, for full training and fine-tuning. I use cross-entropy as the loss function in training the network. For the weighted cross-entropy in the proposed method, the values of C_p and C_n are set to 0.95 and 0.05, respectively. These values are determined based on the number of pixels with and without defects in the training data.

7.3.2 Experimental results

Table 7.1 shows the defect detection accuracy for each method. The experimental results suggest that semantic segmentation is also effective in detecting defects in subway tunnel images. On the other hand, the results obtained in this experiment are not sufficient for practical use, and further improvement of the accuracy should be considered. From the experimental results, the following conclusions can be drawn: From the comparison of the results of FCN full-training and U-Net, the effectiveness of the encoder-decoder structure, especially the symmetric Encoder-Decoder structure, for detecting defects in subway tunnels is confirmed.

Comparing the results of FCN full-training and FCN fine-tuning, the fine-tuning network is

Table 7.1: Result of each semantic segmentation methods.

Method	Recall	Precision	F-value	IoU
FCN full-training [13]	0.347	0.530	0.419	0.220
FCN fine-tuning [13]	**0.495**	0.315	0.385	0.239
U-Net [15]	0.419	**0.592**	**0.491**	**0.325**
ResU-Net [51]	0.209	0.480	0.291	0.173
SegNet [14]	0.092	0.553	0.158	0.113
DeepLab v3+ [19]	0.344	0.383	0.310	0.184

also effective for the defect detection task of subway tunnels by using general images. From the comparison of the results of U-Net, SegNet and FCN full-training, the feature fusion of encoder and decoder is more effective than that of encoder and decoder connected in series for the defect detection task of subway tunnels. DeepLab v3+ has the best detection accuracy for the region segmentation task of general images, but the effectiveness of DeepLab v3+ has not been confirmed in this experiment. Optimization and modification of the network is needed for the region segmentation task of tunnel images. From the experimental results of U-Net and ResU-Net, the method of applying Residual block in U-Net is not effective for the defect detection task in the subway tunnel. From the experimental results, the validity of the Aspp model of DeepLab v3+ is not confirmed.

In the following, I discuss the experimental results for the method using U-Net, which has the best classification accuracy. Figure 7.1 illustrates the detection results of the proposed method. The number of undetected pixels is larger than the number of false positives when compared to Precision and Recall. Additionally, many undetected defects are visible across a wide area in the visualized images. Conversely, some areas also show undetected defects in narrower regions. The possible causes of the undetected defects include. When the encoder feature map is reduced, the location information is lost. There is an imbalance in the number of pixels of each defect type. When the decoder is up-sampling, there's a loss of detail information combined with the encoder's lower-dimension information.

7.4 Summary

In this chapter, I evaluated various methods alongside the U-Net architecture. My conclusions are as follows. The U-Net model requires further improvement for practical multi-scale object segmentation. There are issues, such as false detection due to foreground and background imbalances. Additionally, the precision in fine object segmentation is suboptimal, leading to reduced overall accuracy.

Chapter 8

Defect Detection of Subway Tunnels Using Advanced U-Net Network

8.1 Introduction

Through the demonstration of the effectiveness of the semantic segmentation network in the previous chapters, I have established the technical route for using the semantic segmentation method for supporting tunnel image defect inspection. Next, I will discuss the relevant problems in the current results.

The main remaining problems discussed in the chapter 7 can be explained as follows.

Problem 1: Subway tunnel images contains a high resolution and limited areas of defects. Hence, the problem of imbalance between the background and foreground in datasets is prominent.

Problem 2: Defects in subway tunnels contains multi-scale variations. It is necessary to distinguish these types since the repair operation is different depending on the defect type.

Problem 3: The dataset contains a large proportion of sub-pixel objects (various cracks). The lack of detection accuracy and misdetection of these objects have a significant impact on the accuracy of the model and will also mislead the inspectors in the actual application process.

Hence, it is desirable to devise more effective network architectures that can recover the details of defects in subway tunnel images and improve the detection accuracy for multi-scale defects. This chapter mainly focuses on Problems 1 and 2.

To solve the above problems, I focus on U-Net [15], one of the most widely used methods in biomedical image segmentation tasks. The skip connection method of U-Net, which can con-

catenate up-sampled feature maps with feature maps skipped from an encoder, makes it possible to capture details and location information about objects effectively. U-Net and its variants have achieved impressive segmentation results in computer vision tasks, particularly in detecting multi-scale objects [53–56]. Since the crack features in this task are long and thin, it is necessary for the network to maintain these features with high resolution. Specifically, small objects (such as cracks) are mainly captured by the high-resolution layers, whereas large objects (such as water leakage) are mostly captured by the low-resolution layers. It is easy to add extra modules or change the architecture to improve the detection capacity for different segmentation objects in this task because of its concise architecture. Thus, U-Net architecture is suitable for this task.

In this chapter, an improved version of the U-Net architecture is proposed. To solve the Problem 1, I adjust the image dataset to balance the background and foreground images to overcome the problem of background examples dominating the gradients. To solve the Problem 2, the network architecture is optimized using the following strategies. First, I replace all convolution blocks of the U-Net architecture with inception blocks [57]. Because the inception module consists of four different branches with different kernel sizes and enlarges the receptive field of the network, it can improve network adaption to different scales of features. This improvement increases the capacity to detect multi-scale defects. In addition, for the same purpose, the first convolution layer of the bridge layer is replaced with an atrous spatial pyramid pooling (ASPP) module from DeepLab v2 [18]. It can realize more precise detection and mitigate the overfitting problem by combining these structures.

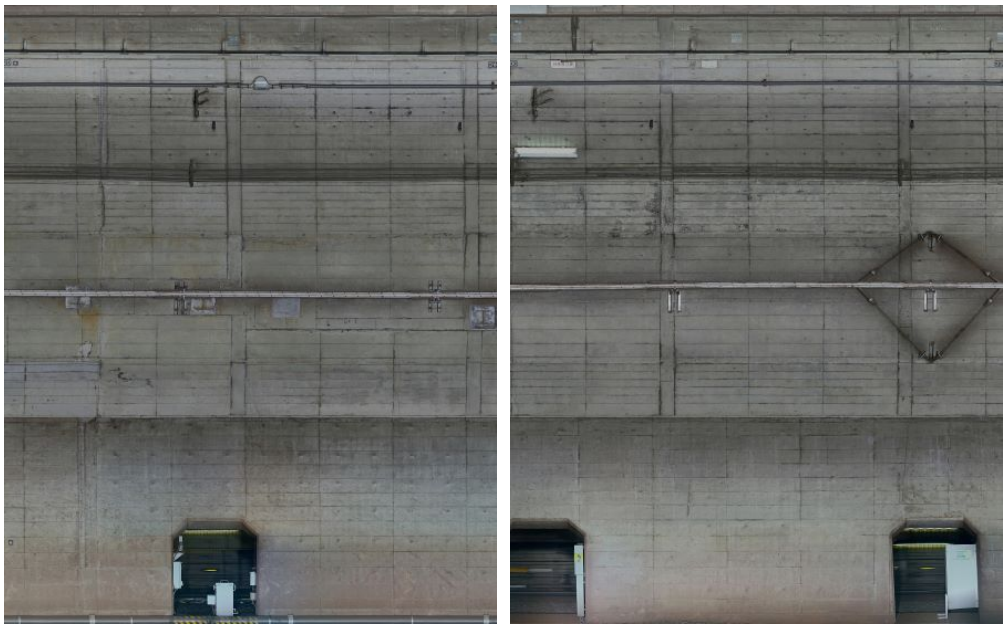
Contributions of this chapter are summarized as follows.

- I propose an advanced U-Net method for defect detection using subway tunnel images.
- I design an architecture that can capture the characteristics of various defects. The experimental results show the effectiveness of the newly proposed architecture.

8.2 Related works

In this section, related works on the U-Net family is discussed.

In 2015, the U-Net architecture was proposed. As a well-known biomedical image segmentation network, the U-Net features a completely symmetric encoder-decoder structure, U-Net ex-



(a) (b)

Figure 8.1: Examples of subway tunnel images used in this study. (resolution: 1 mm/pixel, image size: $12,088 \times 10,000$ pixels).

tracts features from the same size convolutional layers and concatenated with corresponding up-sampling layers; thus, high-level or low-level feature maps can be preserved and inherited by the decoder to obtain more precise segmentation accuracy. After that, its variants were proposed in the following years and are still applied to real-world segmentation tasks today.

Commonly improved variants of U-Net focus on redesigning convolutional modules and modifying down- and up-sampling. Specifically, various methods such as TernausNet [58], Res-UNet [59], Dense U-Net [60], and R2U-Net [55] have been proposed. For example, TenausNet replaces the encoder part with VGG11, Res-UNet and Dense U-Net replace all submodules with residual-connection and dense-connection modules, and R2U-Net combines recurrent convolution and res-connection as a submodule. U-Net++ [53] and U-Net 3+ [54] hope to increase multi-scale target detection capacity. The main advantage of these variants is that, with different receptive fields, they can capture features at varying scales, enabling them to adapt more effectively to object variations at different scales.

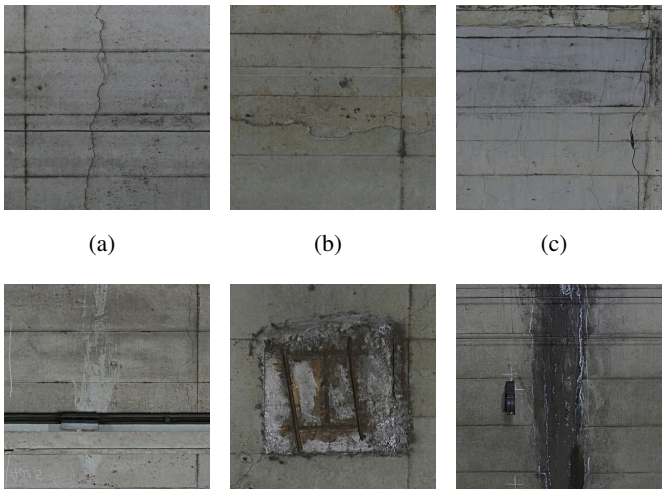


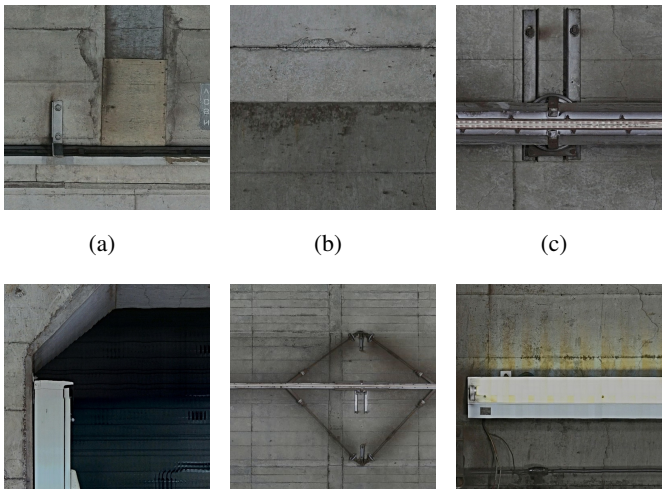
Figure 8.2: Example of defect images. (a–f) represent cracks, cold joint, construction repair, deposition, peeling, and trace of water leakage, respectively. (resolution: 1 mm/pixel, image size: 256×256 pixels).

8.3 Dataset

In this section, I explain the inspection data used in my study. Fig. 8.1 shows examples of the subway tunnel image data. The size of the high resolution images is approximately $12,088 \times 10,000$ pixels or $12,588 \times 10,000$ pixels. With 1mm/pixel resolution, these images can be considered as high-resolution images. Typically, analyzing high-resolution images requires enormous computer resources, and such image sizes are not used in the input of deep learning models. On the other hand, the resizing process results in the loss of fine-scale defects. I solve this problem by the patch division processing.

Fig. 8.2 shows defect patch examples divided from original images ((a) cracks, (b) cold joint, (c) construction repair, (d) deposition, (e) peeling, and (f) trace of water leakage). As shown in Fig. 8.2, each type of defect has its own characteristics, such as different texture edges and color features. As for a two-class segmentation task, this intra-class variance will cause false alarms. For instance, the size and color of cracks (Fig. 8.2 (a)) are different from those of traces of water leakage (Fig. 8.2 (f)).

Next, in Fig. 8.3, I show divided patch examples of background images that have no defects: (a) cable, (b) concrete joint, (c) connection component of overhead conductor rail, (d) passage



(a) (b) (c)

(d) (e) (f)

Figure 8.3: Example of background images. (a–f) show cable, concrete joint, connection component of overhead conductor rail, passage tunnels, overhead conductor rail, and lighter, respectively. (resolution: 1 mm/pixel, image size: 256×256 pixels).

tunnels, (e) overhead conductor rail, and (f) lighter. In Fig. 8.3, some of them have characteristics similar to those of defect images, which can also cause a serious false alarm problem.

8.4 Methodology

Inspired by Inception-v4, ASPP module, and U-Net, I propose a new model for defect detection. The proposed network combines the advantages of the all three existing models. I explain data augmentation in Section 8.4.1 and introduce the architecture of the network in Section 8.4.2.

8.4.1 Data augmentation

In this subsection, I propose the data augmentation strategy and patch selection method. First, I divide high-resolution subway tunnel images into multiple patches as shown in Fig. 8.2 and Fig. 8.3. Let $P_i(i = 1, 2, ..., I)$ denote divided patches derived from the original images shown in Fig. 8.1, where I represents the number of patches. Because of the imbalanced distribution and multi-scale defects, I used an overlapping strategy to ensure exhaustive coverage of defect patches, extending the patch dataset. In addition, to construct the dataset via patch selection, I experimentally obtained a large-scale dataset containing background B_n $(n = 1, 2, ..., N)$ and

defect patches D_m ($m = 1, 2, ..., M$). Note that the ratio between M and N is approximately 7:3 and $N + M = I$.

For the training phase, since the dataset includes superfluous patches and approximately half of them are background patches, this can cause a data imbalance problem. Under this condition, I randomly excluded some background patches to balance the number of patch samples. It should be noted that this strategy does not influence the detection accuracy. Finally, the ratio between defect and background patches can reach 1:1.

The advantage of data augmentation is that features between data distributions can be resolved by pseudo-data generation. The model acquires a high degree of generality by learning to identify the transformed images as the input. In recent years, this idea has been incorporated into self-supervised learning. In self-supervised learning, a transformation similar to data augmentation is performed, and learning is performed without labels. It has been reported that this method can dramatically improve the representative capability of the model itself.

8.4.2 Network architecture

In this subsection, I explain the network architecture used in the proposed method. Fig. 8.4 illustrates the model architecture of the proposed method, while Table 8.1 provides the detailed specifications of the network. I adapt U-Net as the backbone model to achieve optimal performance in the specialized data segmentation task. To increase the rate of detection of multi-scale defects in subway tunnel data, first, I replace the convolution blocks of the U-Net architecture with inception blocks modified from Inception-v3 as shown in Table 8.1. The Inception blocks can enhance the feature capture areas, thereby improving accuracy and mitigating the risk of over-fitting. Second, I add the ASPP module to my model, and I imitate the usage of the ASPP in DeepLab v3+ to set it after the last layer of the encoder (the bridge layer, middle of the network) shown in Fig. 8.5 (a). In more shallow architectures, the final layer of the encoder typically has a size not smaller than 16×16. I modify the parameter settings for the multiple parallel atrous convolutions in the ASPP module to better adapt to the specific task.

The proposed network comprises stacked layers of modified inception blocks, as illustrated in Fig. 8.5 (b), within the U-Net-based encoder-decoder framework. The inception blocks consist of four parallel branches. Three of these branches feature convolution layers with varying kernel

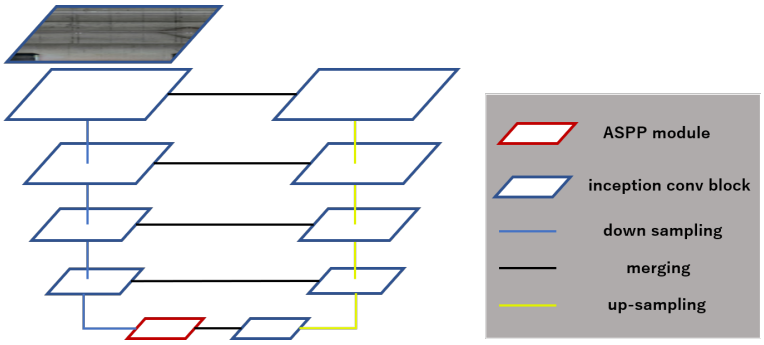


Figure 8.4: Overview of the defect detection network architecture.

sizes, while the last branch contains a max-pooling layer. To reduce the number of training parameters, I replace the 5×5 convolution layer with sequential 5×1 and 1×5 convolution layers. In the original U-Net architecture, the encoder part contains 8 convolution blocks. In addition, the output of every 2 convolution blocks is down-sampled by a max-pooling layer, and to construct a deeper network, I add one inception block before each max-pooling layer, increasing the total number of convolution operations in the encoder from 8 to 12.

At the conclusion of the encoder, the initial convolution layer of the bridge is substituted with an ASPP module. As shown in Fig. 8.5 (a), the input is partitioned into five equal segments. In the original ASPP module, the atrous rates of three 3×3 convolutions are set to 6, 12, and 18 (with 256 filters and batch normalization) to accommodate an input size exceeding 37×37. When the rate value is close to the feature map size, the 3×3 filter degenerates to a 1×1 filter, and atrous convolution loses its effectiveness. For a specific task, given that the input size is restricted to 256×256 pixels, and after 4 max-pooling operations, the final input size of the ASPP module is 16×16, which is less than the required 37×37. Therefore, I changed the atrous rates from 4, 8, and 16 to 2, 4, and 6, respectively, to adapt to the input size. After the ASPP module, a 1×1 convolution operation (with 1,024 channels) is added to merge the bridge layer.

In the decoder part, I used a convolution transpose layer (with a kernel size of 3×3 and a stride size of 2) to perform the up-sampling operation. Instead of using a deeper architecture as the encoder, all basic convolution layers are replaced with inception blocks.

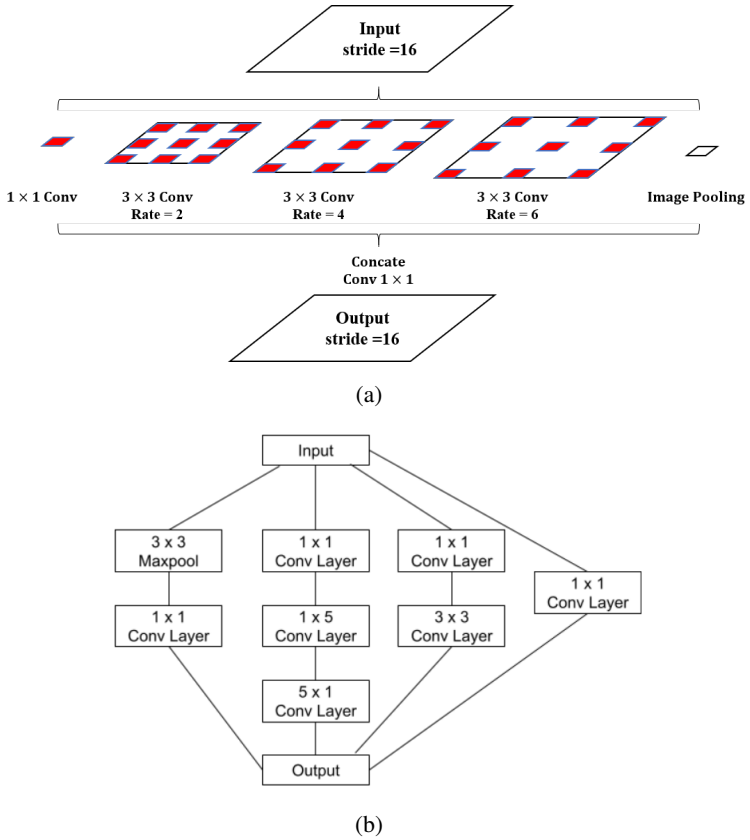


Figure 8.5: Modules introduced in my method. (a) represents the architecture of ASPP module and (b) represents the inception module.

8.5 Experiments and results

This section presents quantitative and qualitative evaluations to confirm the effectiveness of the network for detecting defects in subway tunnel images. The experimental settings are explained in Section 8.5.1 and the results and discussion are presented in Sections 8.5.2 and 8.5.3, respectively. Experimental data are provided by Tokyo Metro Co., Ltd, a Japanese subway company.

8.5.1 Settings

The subway tunnel image dataset consisted of 47 images. The images are obtained from visible cameras with high resolution (e.g., $12,088 \times 10,000$ pixels or $12,588 \times 10,000$ pixels), and the images are divided into multiple patches of 256×256 pixels with a sliding interval of 64 pixels.

The inspectors determined the pixel-wise ground truth for the defects. I selected 280,000

patches from 29 images as the training dataset. In this dataset, the ratio of background to defect patches is set to 1:1. In the validation phase, seven images are divided using the same strategy as in the training phase, and finally, 71,818 patches are selected. The remaining 11 images are used in the test phase. I used the same dividing strategy without abandoning the background patches. Therefore, the number of patches used in the test phase is 326,172, which is significantly larger than that in the training phase. After the test phase, estimated images are generated by recombining the estimated results with the average probability of each pixel.

For the semantic segmentation task, Recall, Precision, F-measure, and Intersection over Union (IoU) are used to evaluate the binary classification performance as my estimation metrics. They can be calculated as follows.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8.2}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{8.3}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{8.4}$$

where TP, TN, FP, and FN represent the number of true-positive, true-negative, false-positive, and false-negative samples, respectively.

I compared the proposed method with classic segmentation methods including DeepLab v3+ (CM1) [19], FCN (CM2) [13], and SegNet (CM3) [14]. Since the input of the network is set to 256×256 pixels, the output size of the encoder in DeepLab v3+ is 16×16. According to the proposed method, I adjusted the parameter settings of multiple parallel atrous convolutions in the ASPP module using the same strategy as introduced in Section 8.4.1. In addition, since the proposed network is based on the U-Net architecture, I also included several earlier U-Net versions as comparative methods (CM4-CM7). The design of each method is shown in Table 8.2. Among them, CM5 [61] added additional down-sampling blocks to both the encoder and decoder of the network, changing the down-sampling stride from 16 to 32.

8.5.2 Results

In this subsection, I show the evaluation results and discuss some important details of the proposed model.

Quantitative analysis

Table 8.3 shows the detection rate of all defects. Among these metrics, IoU, the standard semantic segmentation field metric, is the most important value for evaluating the total performance. It is evident that the proposed method (PM) significantly outperformed all comparative methods (CMs) in this metric.

Table 8.4 lists the recall rate of detection for each defect. It should be noted that the metric recall is used to evaluate each defect detection performance because small crack defects are directly included. For the evaluation of the detection performance of cracks, the IoU is not the best evaluation metric because of the difficulty of pixel-level matching. Moreover, considering the application situation, over-detection is preferable to misdetection to detect defects. For these reasons, I chose the recall metric for this evaluation.

The proposed method outperforms all comparative methods. According to Table 8.3 and Table 8.4, I can further discuss the importance of each component.

Limitation of DeepLab v3+ (CM1):

DeepLab v3+ used atrous convolution, ASPP module, and a simplified decoder branch, achieving great improvement compared with the baseline. There is a slight difference in the detection accuracy for various defects. While these methods can maintain detection accuracy for small objects such as cracks, they show insufficient detection capacity for large objects. as shown in Table 8.4.

FCN and SegNet (CM2, CM3):

FCN and SegNet, as classic segmentation networks, show a certain degree of incompatibility in the subway tunnel dataset, not only with low accuracy but also with a large number of false detection instances as shown in Table 8.3. In particular, the performance of SegNet is extremely poor. Although the detection accuracy of small objects, such as cracks, can

be maintained, it is almost impossible to detect large defects as shown in Table 8.4. This results in low overall detection accuracy and network precision. Unlike U-Net, the SegNet decoder uses the max-pooling indices from the corresponding encoder to perform nonlinear upsampling of the input feature map as a typical symmetric encoder-decoder architecture. It is considered that this function does not perform well in subway tunnel datasets.

Effectiveness of ASPP module (CM4):

In CM4, this module increased the F-measure from 0.428 to 0.444 and IoU from 0.272 to 0.286 compared with the baseline module (CM7) in Table 8.3. Additionally, the results obtained from Table 8.4 suggest that adding the ASPP module significantly improves the detection performance for small-scale and large-scale defects. The results demonstrate the effectiveness of the ASPP module.

Effectiveness of layer extend operation (CM5):

In CM5, compared with the baseline (CM7), this module increases the F-measure from 0.428 to 0.495 and IoU from 0.272 to 0.329, as shown in Table 8.3. Additionally, Table 8.4 suggests that CM5 is superior to CM6 and the baseline (CM7). These results suggest that deeper networks improve the detection of all the defect scales. However, this operation cannot be applied to networks with the ASPP module due to patch size limitations in the experimental setting.

Effectiveness of Inception module (CM6):

In CM6, all the convolution blocks are replaced with the inception module. This operation increased the F-measure from 0.428 to 0.443 and IoU from 0.272 to 0.285 compared to the baseline (CM7) in Table 8.3. Additionally, Table 8.4 shows that the detection rate of each scale significantly improved compared with the baseline. This indicates that adding the inception module can contribute to the representation ability of low-level and high-level information.

Analysis of the proposed method:

As shown in Table 8.3, PM outperformed all other methods. Furthermore, from Table 8.4, it can be seen that PM achieves better accuracy in detecting large-scale defects but has

some limitations in detecting small-scale defects. The limitations of small-scale defects may influence the detection performance of inspection tasks. Thus, qualitative analysis is required.

Qualitative analysis

In this part, I discuss the visual quality of the results. The estimation results are shown in Figs. 8.6-8.9. Fig. 8.6 shows detection result samples of all regions of the test image. Fig. 8.7 and 8.8 show the detection results of peeling and cracks. From Figs. 8.6-8.8, I can see that PM achieves a high detection quality when detecting various defects compared to CMs. On the other hand, Fig. 8.9 displays a sample result of over-fitting. In some cases, I observed that vertical cracks tend to over-fit in my model. The quantitative analyses indicate that the proposed method has some limitations in detecting small-scale defects. However, as Fig. 8.8 suggests, these limitations might not affect actual inspection tasks. Compared to all CMs, the results from PM exhibit fewer instances of false detection, potentially reducing unnecessary work for inspectors.

8.5.3 Discussion

Various models have been proposed for image recognition owing to the AI boom. For general object recognition models, the recognition error rate surpasses that of human capabilities, suggesting a shift towards more advanced tasks. AI applications are beginning to be explored in all areas, including infrastructure maintenance. In this chapter, a method for detecting defects in subway tunnel images is proposed. By constructing a model that considers the characteristics of the data, the proposed method achieved a higher accuracy in detecting defects than conventional methods.

A key consideration is the level of accuracy that the system must achieve to be applicable in real-world scenarios. The quantitative evaluation results obtained from this experiment showed that the IoU is approximately 0.3-0.4. This value may not be sufficient compared to the accuracy of general image recognition. However, as shown in the qualitative evaluation results, cracks and other defects in the image can be detected even if there is some deviation. Considering the practical applications of the proposed method, such as aiding in defect registration for CAD systems or pinpointing defect-dense regions, it can be concluded that the method is suitable for practical implementation.

There are some limitations in this study. This study uses data from the subway line in Japan, and there is still room for future studies on its general applicability to a wide variety of data. In this study, 47 high-resolution subway tunnel images are divided into patches for network training. However, it would be desirable to have a larger number of images to verify the robustness of the proposed method. In addition, since the accuracy is considered to vary depending on the year of tunnel construction, verification using a wide variety of data is necessary. Specifically, the condition of the wall depends on the construction method of the subway tunnel. Furthermore, the new construction method may be completely different from the conventional construction method. When considering the versatility of a model, it is necessary to verify its applicability to various types of data.

8.6 Conclusions

In this chapter, I present a new version of the U-Net architecture to improve defect detection performance in subway tunnel images. By introducing ASPP and inception modules in the U-Net-based network architecture, the capacity of the network for defect detection is improved. The experimental results on a real-world subway tunnel image dataset showed that the proposed method outperformed the other segmentation methods quantitatively and qualitatively. Different from conventional crack-detection methods, the proposed model can detect various types of defects in a single model, which enhances the practicality of supporting tunnel inspections. In future work, I will investigate a new strategy for enhancing the detection accuracy and discuss its application to other real-world applications.

Table 8.1: Architecture of the proposed model.

Type	**Size/Stride**	**Output Size**	**Depth**
Inception Module	$3 \times 3/1$	$256 \times 256 \times 64$	3
Inception Module	$3 \times 3/1$	$256 \times 256 \times 64$	3
Inception Module	$3 \times 3/1$	$256 \times 256 \times 64$	3
Max Pooling	$3 \times 3/2$	$128 \times 128 \times 64$	1
Inception Module	$3 \times 3/1$	$128 \times 128 \times 128$	3
Inception Module	$3 \times 3/1$	$128 \times 128 \times 128$	3
Inception Module	$3 \times 3/1$	$128 \times 128 \times 128$	3
Max Pooling	$3 \times 3/2$	$64 \times 64 \times 128$	1
Inception Module	$3 \times 3/1$	$64 \times 64 \times 256$	3
Inception Module	$3 \times 3/1$	$64 \times 64 \times 256$	3
Inception Module	$3 \times 3/1$	$64 \times 64 \times 256$	3
Max Pooling	$3 \times 3/2$	$32 \times 32 \times 256$	1
Inception Module	$3 \times 3/1$	$32 \times 32 \times 512$	3
Inception Module	$3 \times 3/1$	$32 \times 32 \times 512$	3
Inception Module	$3 \times 3/1$	$32 \times 32 \times 512$	3
Max Pooling	$3 \times 3/2$	$16 \times 16 \times 512$	1
The ASPP module	–	$16 \times 16 \times 1024$	2
Inception Module	$3 \times 3/1$	$16 \times 16 \times 1024$	3
Deconvolution	$3 \times 3/2$	$32 \times 32 \times 512$	3
Cat	–	$32 \times 32 \times 512$	1
Inception Module	$3 \times 3/1$	$32 \times 32 \times 512$	3
Inception Module	$3 \times 3/1$	$32 \times 32 \times 512$	3
Deconvolution	$3 \times 3/2$	$64 \times 64 \times 256$	1
Cat	–	$64 \times 64 \times 512$	1
Inception Module	$3 \times 3/1$	$64 \times 64 \times 256$	3
Inception Module	$3 \times 3/1$	$64 \times 64 \times 256$	3
Deconvolution	$3 \times 3/2$	$128 \times 128 \times 128$	1
Cat	–	$128 \times 128 \times 256$	1
Inception Module	$3 \times 3/1$	$128 \times 128 \times 128$	3
Inception Module	$3 \times 3/1$	$128 \times 128 \times 128$	3
Deconvolution	$3 \times 3/2$	$256 \times 256 \times 64$	1
Cat	–	$256 \times 256 \times 128$	1
Inception Module	$3 \times 3/1$	$256 \times 256 \times 64$	3
Inception Module	$3 \times 3/1$	$256 \times 256 \times 64$	3
Sigmoid	$1 \times 1/1$	$256 \times 256 \times 1$	1

Table 8.2: Differences in the proposed method (PM) and U-Net-based comparative methods (CM4-CM7) used in the experiment.

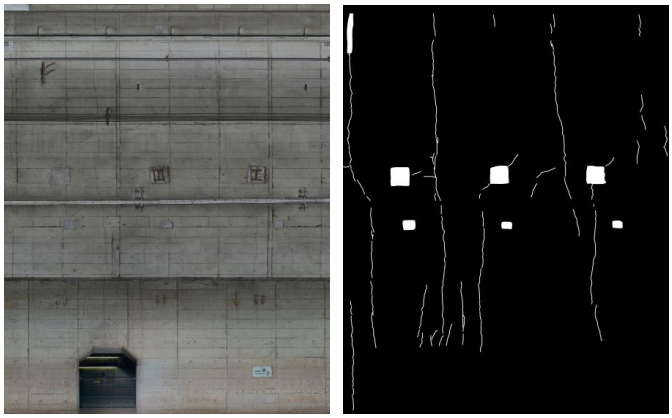
Method	Inception	ASPP	Layer Extend
PM	✓	✓	-
CM4	-	✓	-
CM5	-	-	✓
CM6	✓	-	-
CM7 (Baseline)	-	-	-

Table 8.3: Defect detection performance of the proposed method (PM) and the comparative methods (CMs).

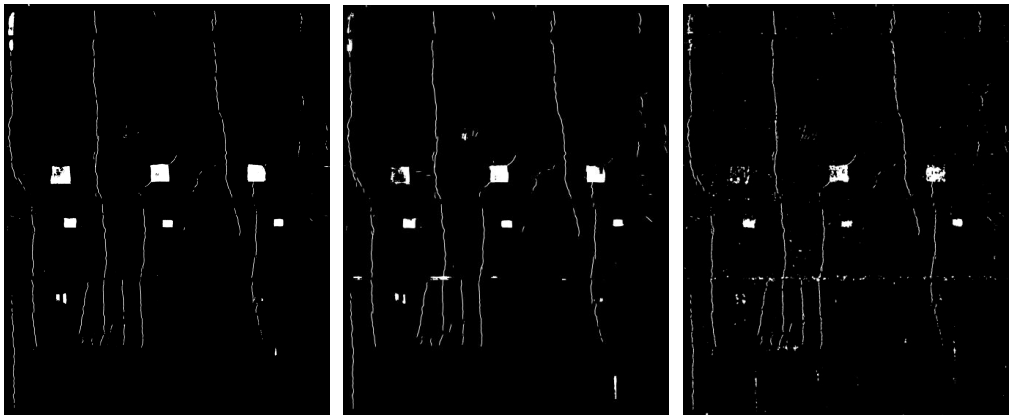
Method	Recall	Precision	F-Measure	IoU
PM	**0.660**	0.436	**0.525**	**0.356**
CM1 [19]	0.564	0.375	0.451	0.291
CM2 [13]	0.494	0.315	0.385	0.238
CM3 [14]	0.410	0.136	0.204	0.158
CM4	0.493	0.405	0.444	0.286
CM5	0.532	**0.463**	0.495	0.329
CM6	0.617	0.346	0.443	0.285
CM7	0.588	0.336	0.428	0.272

Table 8.4: Recall of all kinds of defects in each method.

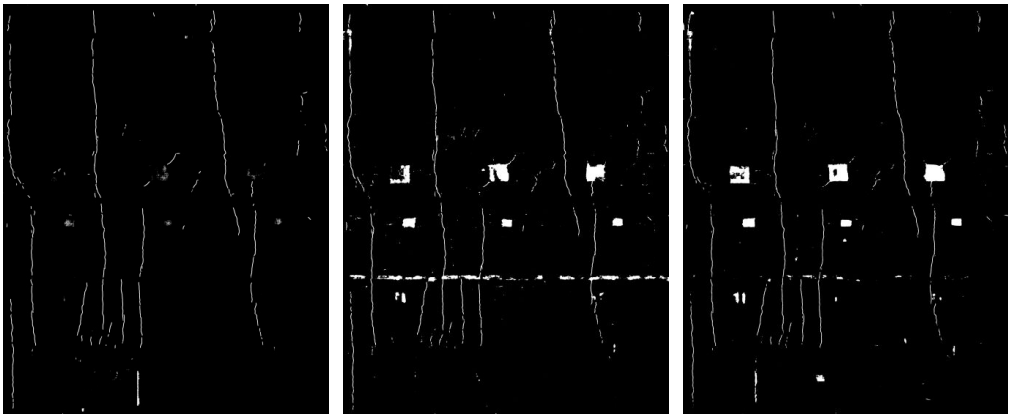
Defect	Recall							
	PM	CM1	CM2	CM3	CM4	CM5	CM6	CM7
Peeling	**0.921**	0.866	0.729	0.191	0.795	0.905	0.711	0.655
Floating	**0.802**	0.711	0.568	0.199	0.708	0.782	0.651	0.533
Crack (0.3 mm-0.5 mm)	0.173	**0.230**	0.163	0.209	0.159	0.140	0.125	0.110
Crack (0.5 mm-1 mm)	0.358	0.385	0.430	0.334	**0.407**	0.382	0.361	0.326
Crack (1 mm-2 mm)	0.402	**0.463**	0.384	0.422	0.455	0.434	0.409	0.388
Crack(2mm+)	0.414	0.409	0.394	0.431	**0.467**	0.444	0.426	0.389
Cold joint	0.013	**0.017**	0.016	0.014	0.016	0.016	0.007	0.005
Honeycomb	0.084	**0.251**	0.230	**0.010**	0.030	0.210	0.090	0.080
Patching (intermediate pile)	**0.819**	0.734	0.616	0.159	0.721	0.816	0.656	0.591
Alligator crack	0.362	0.308	0.216	0.063	0.317	**0.368**	0.306	0.244
Early construction repair	0.423	0.375	0.271	0.061	0.394	**0.504**	0.306	0.297
Deposition	0.054	0.049	0.015	0.001	**0.080**	0.012	0.005	0.010
Construction repair	**0.591**	0.307	0.167	0.078	0.413	0.556	0.364	0.375



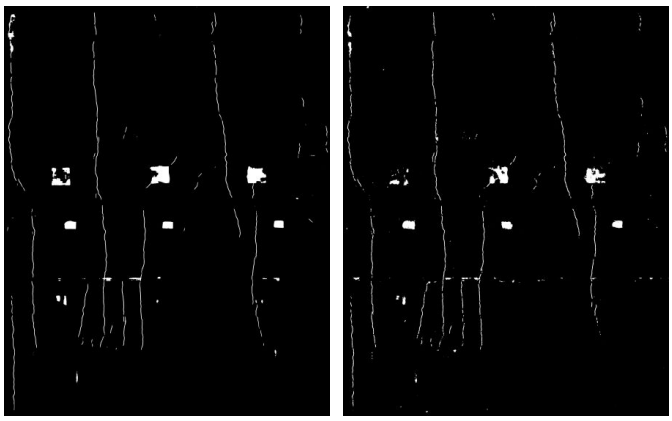
(a) Origin image (b) Ground Truth



(c) PM (d) CM1 (e) CM2

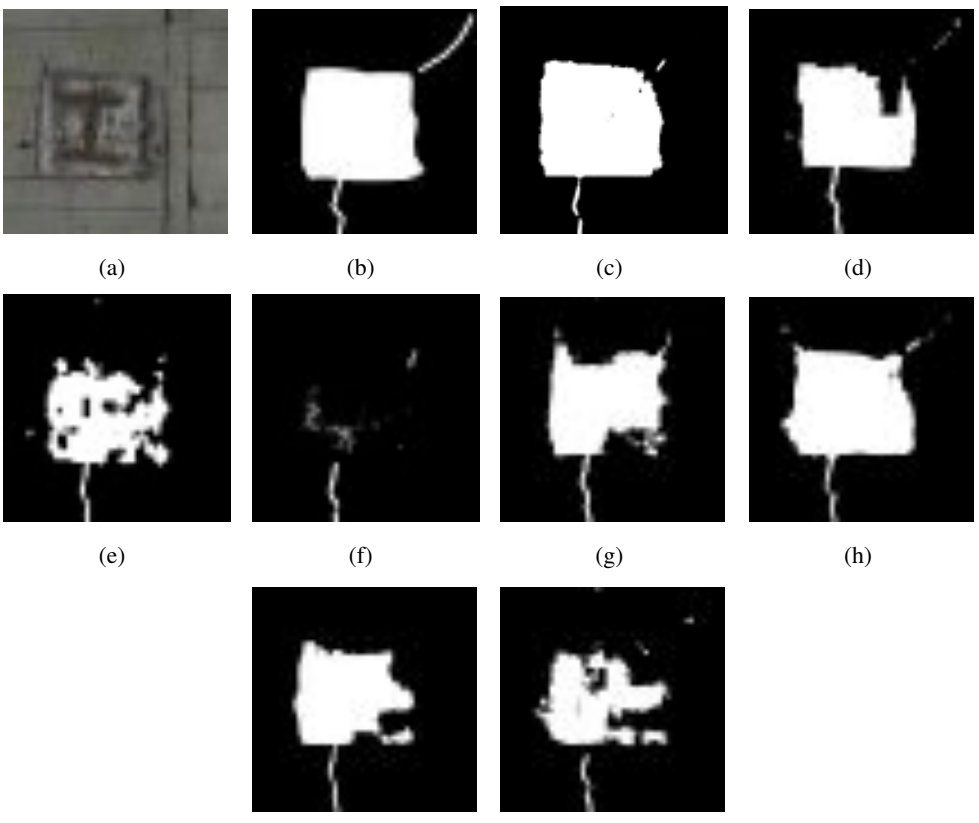


(f) CM3 (g) CM4 (h) CM5



(i) CM6 (j) CM7

Figure 8.6: Results of proposed method and comparative methods. (From left to right: (a): original image; (b): ground truth; (c): results obtained by the proposed method; and (d-j): results obtained by the comparative methods).



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

Figure 8.7: Example of the result in peeling detection. (a) Original image, (b) Ground Truth, (c) PM, (d) CM1, (e) CM2, (f) CM3, (g) CM4, (h) CM5, (I) CM6, (j) CM7.

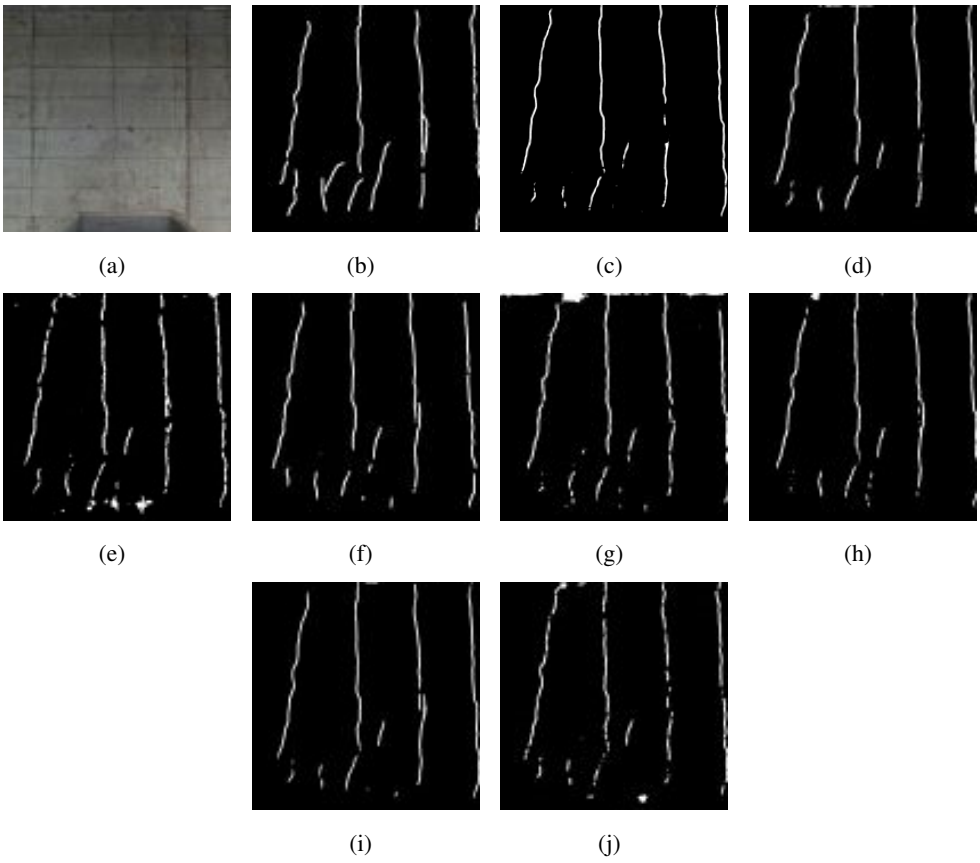


Figure 8.8: Example of the result for crack detection. (a) Original image. (b) Ground truth, (c) PM, (d) CM1, (e) CM2, (f) CM3, (g) CM4, (h) CM5, (I) CM6, (j) CM7

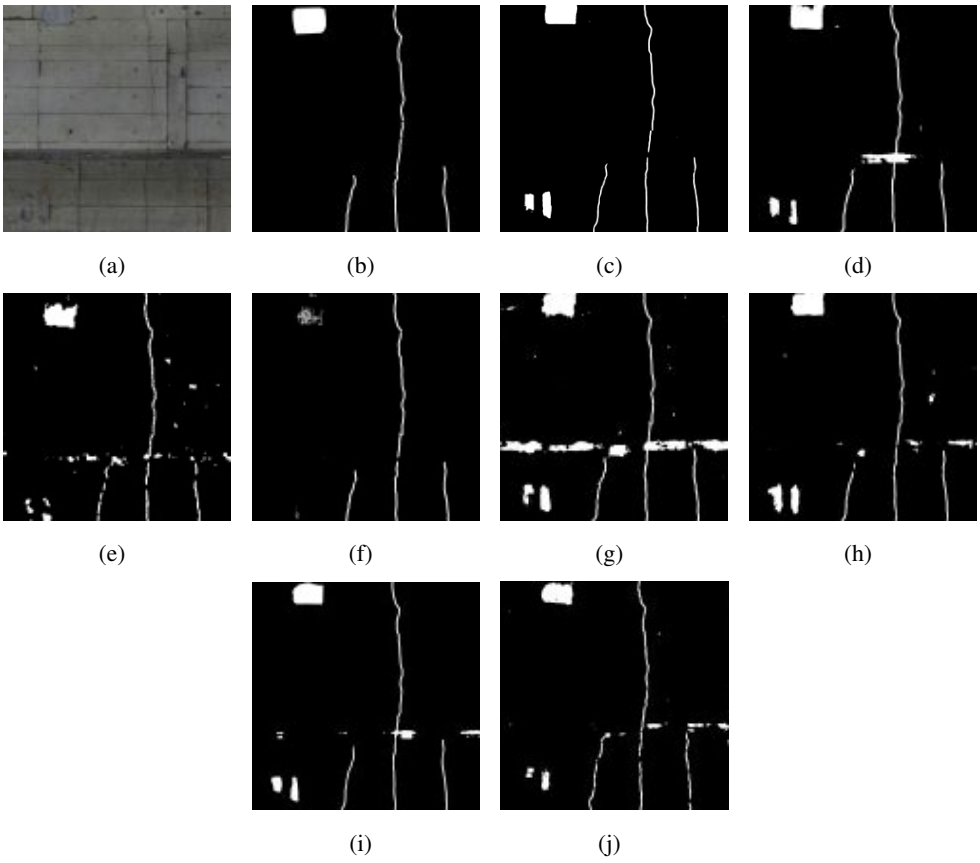


Figure 8.9: Example of the results of over-fitting parts. (a) Origin image, (b) Ground truth, (c) PM, (d) CM1, (e) CM2, (f) CM3, (g) CM4, (h) CM5, (I) CM6, (j) CM7.

Chapter 9

Multi-scale Defect Detection from Subway Tunnel Images with Spatial Attention Mechanism

9.1 Introduction

In the previous chapter, I proposed a defect detection method based on an advanced U-Net [61]. Although this method can achieve good detection accuracy, it still has limitations in its sub-pixel object detection capacity. Specifically, the accuracy of the previous method degrades when the size of the defects is different. Furthermore, because the number of defects varies, bias in the training data is also an issue.

In this chapter, I propose a new defect detection method based on HRNet [21] to solve the above problems. In the proposed method, I adopt a spatial attention (SA) module from DANet [20] and an atrous spatial pyramid pooling (ASPP) module [18] to optimize the foreground detection capacity. Specifically, the multi-scale output of HRNet is optimized by the ASPP module, and then all feature maps are concatenated using the SA module to obtain the estimation result. Through the experiment, the proposed method performs better than some comparative methods.

9.2 Proposed defect detection model

In this section, I explain the network architecture of my defect detection model. First, I use HRNet-W32 as my feature extraction backbone, which is pre-trained by ImageNet. In HRNet, the

Table 9.1: Defect detection performance of the proposed method and the comparative methods.

	ACC			IoU		
	Total	Class1	Class2	MIoU	Class1	Class2
PM	**0.548**	**0.590**	0.375	0.386	**0.448**	0.325
CM1	0.531	0.568	0.380	0.383	0.439	0.328
CM2	0.498	0.535	0.345	0.343	0.382	0.304
CM3	0.468	0.472	**0.414**	**0.392**	0.435	**0.351**

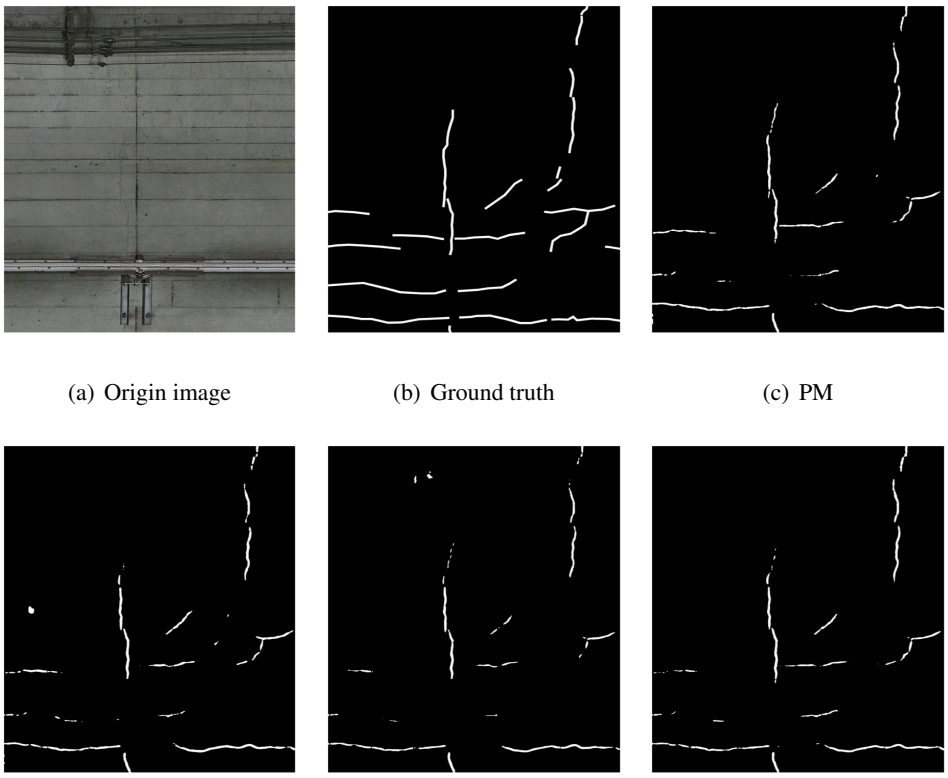
network convolution stream is connected in parallel rather than series. Therefore, the feature map can maintain the characteristics of high resolution. This architecture enables learning the characteristics of all scales and contributes to detecting defects more precisely. Next, these extracted features are optimized using the ASPP module. The ASPP module contributes to expanding the reception field of the convolution network, which can improve the capacity of multi-scale object detection.

Finally, all outputs of the ASPP modules are concatenated using the SA module. The SA module encodes broader contextual information into local features, enhancing its representation capability. Any feature in a particular position is weighted and updated by the features in all positions. The weight is the feature similarity between the two positions. These characteristics can improve the detection capacity of small objects, especially cracks.

9.3 Experiment and conclusion

In this section, I verify the effectiveness of my method through the defect detection experiment. I use 48 subway tunnel images provided by Tokyo Metro Co., Ltd. I treat 35 images for the training data, and the remaining 13 images as test data. Since the tunnel images had high resolutions, I divide them into multiple patches in the size of 512×512 pixels. As a result, the dataset has 33,152 patches for training and 12,684 patches for testing.

The ground truth of the dataset has 18 classes of defects; in my task, I classify them into three classes: “Cracks” as class 1, “Large-scale defects” as class 2, and “Background” as class 3. Note that the background class represents regions without defects. Each class accounts for 10%, 16%, and 74% of the total classes, indicating imbalanced data.



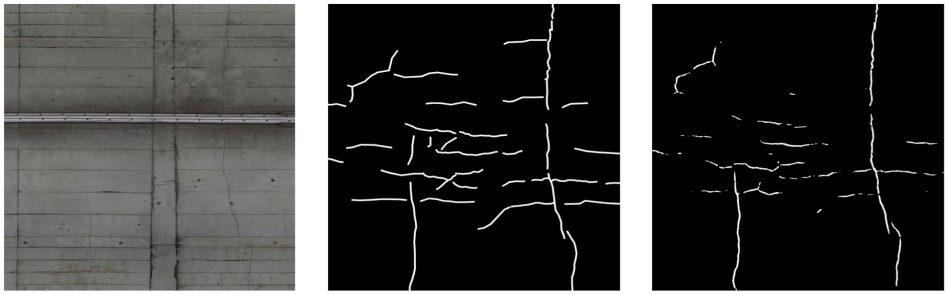
(a) Origin image (b) Ground truth (c) PM

(d) CM1 (e) CM2 (f) CM3

Figure 9.1: Example of the results in crack detection.

Accuracy (ACC) and Intersection over Union (IoU) metrics assess the detection performance. Furthermore, I compare my method with other comparative methods (CM), including CM1 (adopting the ASPP module only), CM2 (adopting the SA module only), and CM3 (baseline HRNet-W32 fine-tuned by my dataset).

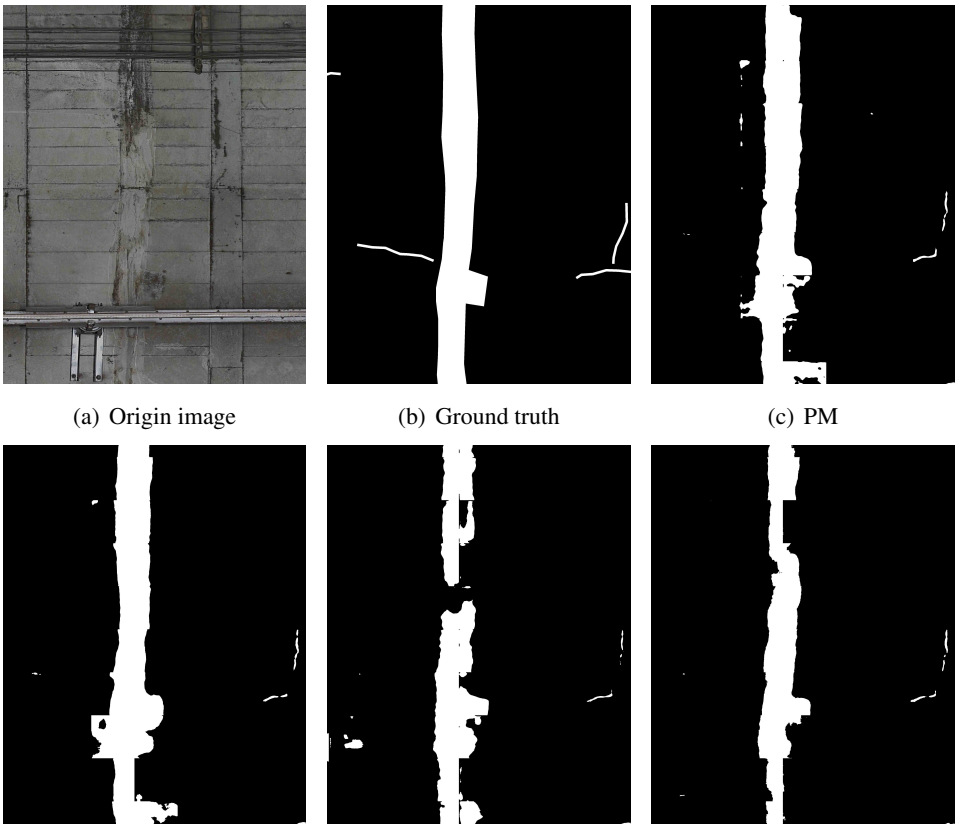
The defect-detection performance of each method is listed in Table 9.1. According to this table, the proposed method achieves a better or comparable performance than CMs 1-3. Next, the proposed network outperforms some methods for detecting class 1 (cracks) as shown in Table 9.1, Fig. 9.1 and Fig. 9.2. However, our method is inferior to the comparative methods in detecting class 2. Pixels belonging to class 2 have missing detection and over-detection problems, as shown in Fig. 9.3. According to Fig. 9.4, false detection is likely to occur when the background features share specific characteristics with the defect features.



(a) Origin image (b) Ground truth (c) PM

(d) CM1 (e) CM2 (f) CM3

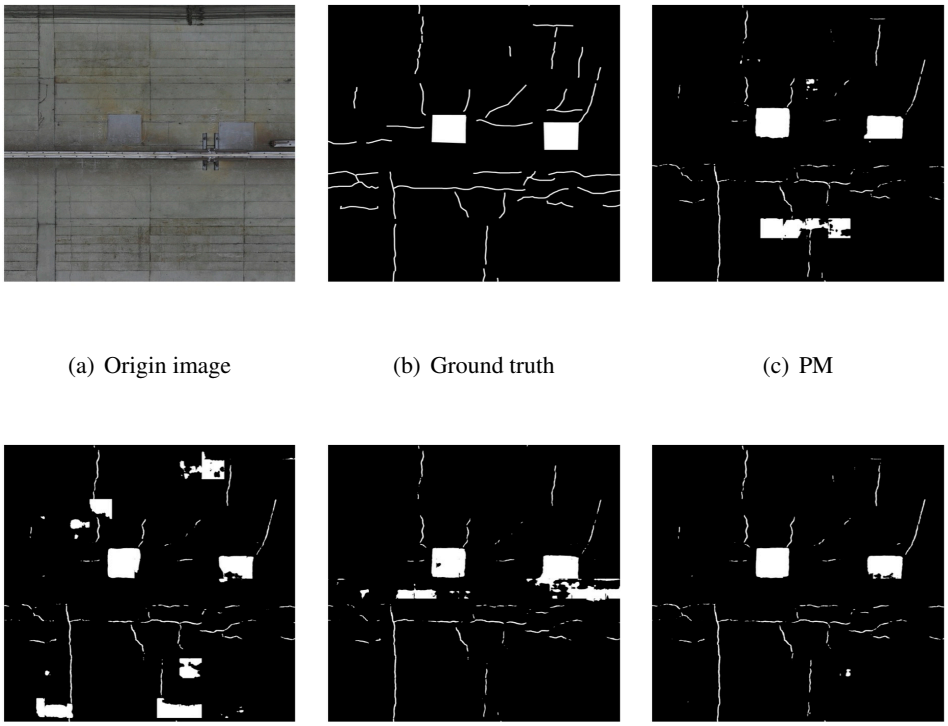
Figure 9.2: Example of the results in crack (0.3mm - 0.5mm) detection.



(a) Origin image (b) Ground truth (c) PM

(d) CM1 (e) CM2 (f) CM3

Figure 9.3: Example of the results in water leakage detection.



(a) Origin image (b) Ground truth (c) PM

(d) CM1 (e) CM2 (f) CM3

Figure 9.4: Example of multi-scale defect detection results.

Chapter 10

Summary

This chapter reviews the contribution and shows future tasks.

10.1 Overview of this thesis

This section provides a proposition overview of this thesis based on the background and previous sections. This thesis aims to develop an accurate defect detection method for supporting infrastructure inspection. To achieve this goal, this thesis carries out the following explorations. First, I continue constructing a defect classification model based on CNN based on the experiment of previous research. As a preparatory experiment, I also carry out preparatory experiments on the semantic segmentation architecture, FCN. Due to the different evaluation systems, despite proving the effectiveness of the classification model of the CNN, particularly the accuracy on smaller targets, there are many problems with the FCN part. Therefore, in Chapter 5, I redesign the experiments of the FCN part and attempt to combine the advantages of CNN and FCN to build a composite model. In this attempt, the model achieve good results; however, the CNN part of the training is more expensive, and the displayed results are not intuitive enough, so for more accessible application in inspection operation, I choose to adopt the FCN architecture only in the next stage. Therefore, in Chapters 6 and 7, I mainly focus on the initial improvement of the U-Net architecture, explore the performance of the semantic segmentation model on infrastructure datasets, and summarize its features and issues. In Chapters 8 and 9, I identify three major problems with the current subway tunnel dataset, which are also the major problems most infrastructure datasets face, and improve the network architecture to solve these problems. 1. foreground-background

imbalance, 2. multi-scale defects, and 3. sub-pixel object. Below, the overview of each chapter of this thesis is reviewed.

Chapter 1 introduces the research background and objectives. Chapter 2 provides the related works of my thesis. Chapter 3 presents an overview of the dataset used in this study, highlighting its key characteristics and information. In Chapter 4, the CNN-based defect detection method is presented. Chapter 5 combines FCN and CNN methods for detecting defects from high-resolution subway tunnel images. In Chapter 6, I discuss the limitations of CNN and FCN in defect detection using tunnel images and propose a new U-Net-based defect detection method. In Chapter 7, I compare the constructed U-Net-based defect detection method with various semantic segmentation methods and confirm the effectiveness of my approach. In Chapter 8, I propose an improved version of U-Net to enhance the defect detection method's capability. Through the experiments, the proposed method demonstrates its performance in identifying issues such as foreground-background imbalance problems and the capacity to detect multi-scale objects during the application process. In Chapter 9, I propose a new HRNet-based network architecture to enhance the robustness of the defect detection method. Through these modifications, I confirm the efficacy of the proposed enhancement approach in improving the detection accuracy of sub-pixel objects.

In this thesis, I use the tunnel structure with a more complicated situation in the concrete structure as the basis to conduct experiments and verifications. Since types of defects in the concrete structures are similar, the methods of this paper can still be applied to the defect detection of various infrastructure surfaces, such as residential walls and dam surfaces. And it can also be applied to road defect detection, such as highway and airport runways.

10.2 Further tasks

In this thesis, I present defect detection methods to support infrastructure maintenance and evaluate it using infrastructure image datasets. However, in this task, I have enough labeled image datasets to support full-supervised learning, and even in the case of sufficient data, the detection accuracy of the network is still insufficient. When applying to other infrastructure maintenance tasks, I will inevitably face datasets with few labeled and even unlabeled. The current methods

are not enough to deal with this situation. These will be topics I will discuss in the future, namely semi-supervised or unsupervised segmentation of multi-scale infrastructure defects.

Acknowledgements

First, I would like to sincerely thank my supervisors, Prof. Miki Haseyama and Prof. Takahiro Ogawa. This thesis would not have been possible without the invaluable guidance and encouragement they have given me over the 5 years I spent at the Graduate school of Information Science and Technology, Hokkaido University.

Furthermore, I would like to sincerely thank Specially Appointed Assistant Prof. Ren Togo, Specially Appointed Assistant Prof. Keisuke Maeda, and Assistant Prof. Naoki Saito for their constant encouragement and advice about research and academic life. And, I would also like to thank Specially Appointed Prof. Kenji Araki, Specially Appointed Prof. Yuji Sakamoto, and Prof. Yoshinori Dobashi for providing insightful comments and suggestions about the research I performed at the Graduate School of Information Science and Technology, Hokkaido University. Finally, I would like to sincerely thank everyone at the Laboratory of Media Dynamics, Graduate School of Information Science and Technology, Hokkaido University, and my family for their invaluable support and assistance.

References

[1] T. Ministry ofLand, Infrastructure and Tourism, “White paper on land, infrastructure,transport and tourism in japan 2019,” 2019. https://www.mlit.go.jp/en/statistics/white-paper-mlit-2019.html

[2] 山本努, “東京メトロにおけるトンネルの維持管理と長寿命化への取組み (特集 地下鉄の安全安心を考える),” 日本地下鉄協会報，vol.197，pp.26–30，2013.

[3] S.-N. Yu, J.-H. Jang, and C.-S. Han, “Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel,” Automation in Construction, vol.16, no.3, pp.255–261, 2007.

[4] E. Menendez, J.G. Victores, R. Montero, S. Martínez, and C. Balaguer, “Tunnel structural inspection and assessment using an autonomous robotic system,” Automation in Construction, vol.87, pp.117–126, 2018.

[5] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, “Cracktree: Automatic crack detection from pavement images,” Pattern Recognition Letters, vol.33, no.3, pp.227–238, 2012.

[6] H. Huang, Y. Sun, Y. Xue, and F. Wang, “Inspection equipment study for subway tunnel defects by grey-scale image processing,” Advanced Engineering Informatics, vol.32, pp.188–201, 2017.

[7] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, “Automatic crack detection and classification method for subway tunnel safety monitoring,” Sensors, vol.14, no.10, pp.19307–19328, 2014.

[8] L. Zhang, F. Yang, Y.D. Zhang, and Y.J. Zhu, “Road crack detection using deep convolutional neural network,” in Proceedings of the International Conference on Image Processing (ICIP), pp.3708–3712, 2016.

[9] N.L.D. Khoa, A. Anaissi, and Y. Wang, “Smart infrastructure maintenance using incremental tensor analysis,” in Proceedings of the ACM on Conference on Information and Knowledge Management, pp.959–967, 2017.

[10] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, “Automatic pixel-level crack detection and measurement using fully convolutional network,” Computer-Aided Civil and Infrastructure Engineering, vol.33, no.12, pp.1090–1109, 2018.

[11] J.A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” Neural Processing Letters, vol.9, no.3, pp.293–300, 1999.

[12] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proceedings of the Advances in Neural Information Processing Systems, pp.1097–1105, 2012.

[13] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.3431–3440, 2015.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.39, no.12, pp.2481–2495, 2017.

[15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp.234–241, 2015.

[16] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1925–1934, 2017.

[17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.2881–2890, 2017.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.40, no.4, pp.834–848, 2017.

[19] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proceedings of the European Conference on Computer Vision (ECCV), pp.801–818, 2018.

[20] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.3146–3154, 2019.

[21] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., “Deep high-resolution representation learning for visual recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.43, no.10, pp.3349–3364, 2020.

[22] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” in Proceedings of the Advances in Neural Information Processing Systems, vol.34, pp.12077–12090, 2021.

[23] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the International Conference on Computer Vision (ICCV), pp.10012–10022, 2021.

[24] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for universal image segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1290–1299, 2022.

[25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” arXiv:2304.02643, pp.1–30, 2023.

[26] Y. Xue and Y. Li, “A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects,” Computer-Aided Civil and Infrastructure Engineer-

ing, vol.33, no.8, pp.638–654, 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Proceedings of the Advances in Neural Information Processing Systems, pp.91–99, 2015.

[28] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in Proceedings of the Advances in Neural Information Processing Systems, pp.379–387, 2016.

[29] 石原賢太，高橋　翔，小川貴弘，長谷山美紀，“畳み込みニューラルネットワークを用いた地下鉄トンネルにおける変状検出に関する検討,” 映像情報メディア学会技術報告，vol.41，no.5，pp.81–86，2017.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.

[31] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” Computer-Aided Civil and Infrastructure Engineering, vol.32, no.5, pp.361–378, 2017.

[32] A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in Proceedings of the ACM International Conference on Multimedia, pp.689–692, 2015.

[33] 豊田　陽，原川良介，小川貴弘，長谷山美紀，“レーザーデータを用いた地下鉄トンネル内の変状検出に関する検討 ー 全層畳み込みネットワークを用いた変状領域の可視化 ー,” 映像情報メディア学会技術報告，vol.43，no.5，pp.295–299，2019.

[34] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung, “Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle,” Sensors, vol.18, no.6, p.1881, 2018.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.580–587, 2014.

[36] Z. Huang, H. Fu, W. Chen, J. Zhang, and H. Huang, "Damage detection and quantitative analysis of shield tunnel structure," Automation in Construction, vol.94, pp.303–316, 2018.

[37] S.-N. Yu, J.-H. Jang, and C.-S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," Automation in Construction, vol.16, no.3, pp.255–261, 2007.

[38] MichaelO ' Byrne, F. Schoefs, B. Ghosh, V. Pakrashi, "Texture analysis based damage detection of ageing infrastructural elements," Computer-Aided Civil and Infrastructure Engineering, vol.28, no.3, pp.162–177, 2013.

[39] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol.20, no.3, pp.273–297, 1995.

[40] A. Cord and S. Chambon, "Automatic road defect detection by textural pattern recognition based on adaboost," Computer-Aided Civil and Infrastructure Engineering, vol.27, no.4, pp.244–259, 2012.

[41] E.H. Adelson and J.R. Bergen, "Spatiotemporal energy models for the perception of motion," Josa a, vol.2, no.2, pp.284–299, 1985.

[42] I. Brilakis, S. German, and Z. Zhu, "Visual pattern recognition models for remote sensing of civil infrastructure," Journal of Computing in Civil Engineering, vol.25, no.5, pp.388–393, 2011.

[43] K.C. Wang, Q. Li, and W. Gong, "Wavelet-based pavement distress image edge detection with a trous algorithm," Transportation Research Record, vol.2024, no.1, pp.73–81, 2007.

[44] Z. Qingbo, "Pavement crack detection algorithm based on image processing analysis," in Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol.1, pp.15–18, 2016.

[45] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," Computer-Aided Civil and Infrastructure Engineering, vol.33, no.9, pp.731–747, 2018.

[46] T. Sakagami, “Remote nondestructive evaluation technique using infrared thermography for fatigue cracks in steel bridges,” Fatigue & Fracture of Engineering Materials & Structures, vol.38, no.7, pp.755–779, 2015.

[47] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, “Concrete crack detection by multiple sequential image filtering,” Computer-Aided Civil and Infrastructure Engineering, vol.27, no.1, pp.29–47, 2012.

[48] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv:1409.1556, pp.1–14, 2015.

[49] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol.60, no.6, pp.84–90, 2017.

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–9, 2015.

[51] Z. Zhang, Q. Liu, and Y. Wang, “Road extraction by deep residual u-net,” IEEE Geoscience and Remote Sensing Letters, vol.15, no.5, pp.749–753, 2018.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016.

[53] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” arXiv:1807.10165, pp.1–8, 2018.

[54] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, “Unet 3+: A full-scale connected unet for medical image segmentation,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1055–1059, 2020.

[55] M.Z. Alom, M. Hasan, C. Yakopcic, T.M. Taha, and V.K. Asari, “Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation,” arXiv:802.06955, pp.1–12, 2018.

[56] F.I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, “Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data,” ISPRS Journal of Photogrammetry and Remote Sensing, vol.162, pp.94–114, 2020.

[57] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” arXiv:1602.07261, pp.1–12, 2016.

[58] V. Iglovikov and A. Shvets, “Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation,” arXiv:1801.05746, pp.1–5, 2018.

[59] X. Xiao, S. Lian, Z. Luo, and S. Li, “Weighted res-unet for high-quality retina vessel segmentation,” in Proceedings of the International Conference on Information Technology in Medicine and Education (ITME), pp.327–331, 2018.

[60] S. Guan, A.A. Khan, S. Sikdar, and P.V. Chitnis, “Fully dense UNet for 2-d sparse photoacoustic tomography artifact removal,” IEEE Journal of Biomedical and Health Informatics, vol.24, no.2, pp.568–576, 2020.

[61] A. Wang, R. Togo, T. Ogawa, and M. Haseyama, “Detection of distress region from subway tunnel images via u-net-based deep semantic segmentation,” in Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), pp.766–767, 2019.

Achievements of the Author

(A) Journal Papers

[A-1] A. Wang, R. Togo, T. Ogawa, M. Haseyama, "Defect detection of subway tunnels using advanced U-Net network," *Sensors*, no. 22, 2330, 2022.

(B) International Conferences

[B-1] A. Wang, R. Togo, T. Ogawa, M. Haseyama, "Detection of distress region from subway tunnel images via u-net-based deep semantic segmentation," in *Proc. the IEEE Global Conference on Consumer Electronics (GCCE)*, pp. 792-793, 2019.

[B-2] A. Wang, R. Togo, T. Ogawa, M. Haseyama, "Multi-scale defect detection from subway tunnel images with spatial attention mechanism," in *Proc. the IEEE International Conference on Consumer Electronics- Taiwan (ICCE-TW)*, pp. 305-306, 2022.

(E) Lectures

[C-1] 王安, 原川 良介, 小川 貴弘, 長谷山 美紀, "FCN と CNN を用いた地下鉄トンネルにおける変状検出に関する検討," 電気・情報関係学会北海道支部連合大会講演論文集, pp. 30-31, 2018.

[C-2] 王安, 原川 良介, 小川 貴弘, 長谷山 美紀, "畳み込みニューラルネットワークを用いた地下鉄トンネルにおける変状検出の高精度化に関する一検討," 映像情報メディア学会技術報告, vol. 43, no. 5, pp.121-122, 2019.

[C-3] 王安, 藤後 廉, 小川 貴弘, 長谷山 美紀, "Semantic Segmentation に基づく地下鉄トンネルにおける変状検出に関する検討," 電気・情報関係学会北海道支部連合大会講演論文集, pp. 151-152, 2019.

[**C-4**] 王 安, 藤後 廉, 小川 貴弘, 長谷山 美紀, “A note on detection of distress regions in subway tunnels by using U-net based network,” 映像情報メディア学会技術報告, vol. 44, no. 5, pp.69-72, 2020.