Measurement 178 (2021) 109316

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

# Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN

Yingying Xu [a], Dawei Li [b], Qian Xie [b], Qiaoyun Wu [b], Jun Wang [b,*]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
[b] College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics and Astronautics, China

ARTICLE INFO

*Keywords:*
Leakage
Spalling
Defect detection
Deep learning
Mask R-CNN
Instance segmentation

ABSTRACT

The detection of tunnel surface defects is the very important part to ensure tunnel safety. Traditional tunnel detection mainly relies on naked-eye inspection, which is time-consuming and error-prone. In the past few years, many defect detection methods based on computer vision have been introduced. However, these methods with manual feature extraction do not perform well in detecting tunnel defects due to the complicated background of tunnel surfaces. To address these problems, this paper proposes a novel tunnel defect inspection method based on the Mask R-CNN. To improve the accuracy of the network, we endow it with a path augmentation feature pyramid network (PAFPN) and an edge detection branch. These improvements are easy to implement, with subtle extra memory and computational overhead. In this paper, we perform a detailed study of the PAFPN and the edge detection branch, and the experiment results show their robustness and accuracy in tunnel defect detection and segmentation.

## 1. Introduction

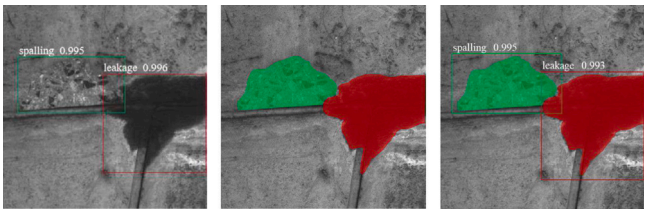
With the rapid development of the urban railway system, about 40 cities' railway systems have been in service by the end of 2020 in China. By the influence of train vibration and building construction, tunnels inevitably decay with their service time increase and surfer from a number of defects. Leakage and spalling are the most common defects existing on tunnel surface [1]. Risk analysis of tunnel lining inspection shows that these defects can seriously threaten the stability and durability of tunnel structure [2,3]. If these defects are not inspected and repaired in time, they would cause significant economic losses and even threaten life safety. For example, the analysis of tunnel accidents shows that tunnel defects, such as leakage, could cause soil settlement, tunnel deformation and serious structural damage, which eventually lead to a series of accidents and the close of the tunnels [4,5]. Therefore, it is very important and urgent-needed to perform tunnel defect inspection and the corresponding maintenance in time to ensure the safety of tunnel operation.

Traditional tunnel surface defect inspection mainly relies on naked-eye inspection, which is carried out by the technical inspectors walking along the tunnel line. This method is inefficient and costly in examining large-area tunnel lining structures. Taking Nanjing metro as an example, there are more than 40 inspectors working in metro tunnels during time window (about 3 h per day) to detect defects in the tunnel surface. However, due to the limited working time, it is difficult to accomplish the inspection task. In addition, such a manual inspection is subjective and prone to erroneous judgements. To address these problems, a number of vision-based techniques have been emerged as more effective methods for tunnel defect inspection.

In the conventional vision-based defect inspection procedure, the images of tunnel surfaces are first collected by image acquisition equipments. Then the handcrafted features are designed for defect detection. German et al. [6] proposed a novel method for automatic spalling detection. At first, a local entropy-based thresholding algorithm was used to segment the spalling regions. Then, the detected defect regions were measured using template matching algorithm and morphological operations. Zhang et al. [7] proposed an automatic defect detection and classification system for tunnel safety monitoring. At first, local dim areas containing potential defects were detected from original images by using morphological operations and thresholding techniques. Following this, a distance-based shape descriptor was designed to describe numerical features for defect detection. Wang et al. [8] designed a two-step method to detect defects on tunnel. In the first step, Otsu threshold technology and Prewitt operator were combined to segment potential defect regions. In the second step, area and perimeter were used to design morphology characteristics for defect feature extraction. In summary, most of the earlier vision-based tunnel defect detection methods mainly rely on improved digital image processing techniques

* Corresponding author.
*E-mail address:* wjun@nuaa.edu.cn (J. Wang).

https://doi.org/10.1016/j.measurement.2021.109316
Received 20 September 2020; Received in revised form 23 February 2021; Accepted 19 March 2021
Available online 27 March 2021




(a) object dectection (b) semantic segmentation (c) instance segmentation

**Fig. 1.** The tasks of object detection, sematic segmentation and instance segmentation task. (a) Object detection task is aim to identify "what" and "where", predicting bounding boxes with corresponding class scores. (b) Sematic segmentation task is aim to identify "where" for pixel level detection. (c) Instance segmentation task is aim to identify "what" and "where" for pixel level detection.

and their combinations, such as edge detection, thresholding and morphological operations. They can achieve good detection results under certain circumstances, but their generalization capability and adaptability are poor. In fact, due to the harsh environment of tunnel lining surfaces, such as insufficient illumination, complicated pipelines and other interferences, it is challenging to construct an efficient defect detection algorithm using conventional image processing technologies.

In recent years, deep learning techniques have brought about breakthroughs in image analysis [9,10]. Researches show that methods based on deep learning techniques are able to learn a hierarchy of semantic features from raw images and adapt to different concrete situations with high robustness and reliability [11]. Karen Simonyan and Andrew Zisserman [12] evaluated very deep convolutional networks for large-scale image classification. The results further confirmed the effectiveness of the depth in visual representation. In the following years, the use of deep learning has shown outstanding performances in many computer vision fields such as object detection [13–16], semantic segmentation [17–19] and instance segmentation [20–23]. Here, we take the tunnel surface defect inspection task as an example. Given a tunnel surface image, the object detection task is to identify the defect categories and locate each defect in the image by predicted bounding boxes with corresponding class scores (as shown in Fig. 1(a)). The semantic segmentation task is to extract defect regions from the image at pixel-level (as shown in Fig. 1(b)). The instance segmentation task is to classify and locate defects in the image and segment each defect at the pixel-level (as shown in Fig. 1(c)).

For the tunnel quality assessment system, the category, quantity, location and geometric information of defects are the crucial indicators to evaluate the risk levels of defects. Therefore, in order to realize a comprehensive information reflection of the tunnel surface, we adopt the instance segmentation method to perform tunnel defect detection and segmentation task. Compared with other methods, the Mask R-CNN algorithm achieves the best performance on the challenging instance segmentation dataset COCO [24]. Therefore, in view of the poor robustness and low recognition of traditional tunnel defect detection algorithms with manual feature extraction, this paper proposes a novel defect inspection method based on the Mask R-CNN [20] to detect and segment leakage and spalling from tunnel surface images.

Mask R-CNN is a state-of-the-art image instance segmentation algorithm, which can learn rich features from input images. However, we find that most of the edge information of tunnel defects is lost using the original Mask R-CNN for tunnel defect inspection. This fact indicates that the feature extraction approach in the original Mask R-CNN can be further improved. It is worth noting that low-level features facilitate the identification of large targets. However, there is a long path from low-level features to top features in the backbone network of Mask R-CNN, leading to the loss of available localization information of low-level features. This problem will affect subsequent task of detection and segmentation of tunnel surface defects. Thus, in order to obtain enhanced features with rich low-level information, we introduce a path augmentation feature pyramid network (PAFPN) which is constructed on the basis of the original Mask R-CNN, so as to boost the information propagation of the feature extraction approach. In fact, the models in [17,18,25,26] [19,27] also use the low-level features. However, they do not make full use of them to enhance the entire feature hierarchy for object recognition.

In the other hand, to further improve the accuracy of the edges of the Mask R-CNN segmentation results, we add an edge detection branch to the end of the network, which enables the network to generate feature maps focusing on edge information. Edge detection has been a research topic for many decades, so there are many traditional methods in this field [28–30]. In recent years, with the development of deep learning techniques, some CNN-based edge detectors have appeared [31–33]. Note that unlike high-resolution color images, the target images in our work are $28 \times 28$ sized with only one channel depicting a single instance. This significantly reduces the complexity of the edge detection task and justifies the rationality of simple edge detection filters. Therefore, we choice the Sobel image gradient filters [28] for this problem because it keeps the computational overhead to a minimum. In addition, the proposed model requires a large number of images for training and testing. However, few metro tunnel surface images containing leakage and spalling have been obtained so far. To solve this problem, we use a self-developed tunnel image acquisition equipment to collect tunnel surface images and the defect areas in images are labeled for database construction.

Another problem for defect detection is that most previous studies only focus on detecting defects, while paying little attention to the defect risk level problem, which is essential for tunnel quality assessment and subsequent repair and protection. In practical, the risk levels of defects are usually estimated by the areas of defects. However, during the database construction process, a complete defect region may be cropped into different images, as demonstrated in Fig. 2 (Stage 1). In order to merge the detected defect regions belonging to the same one, we propose an algorithm to merge each connected defect region.

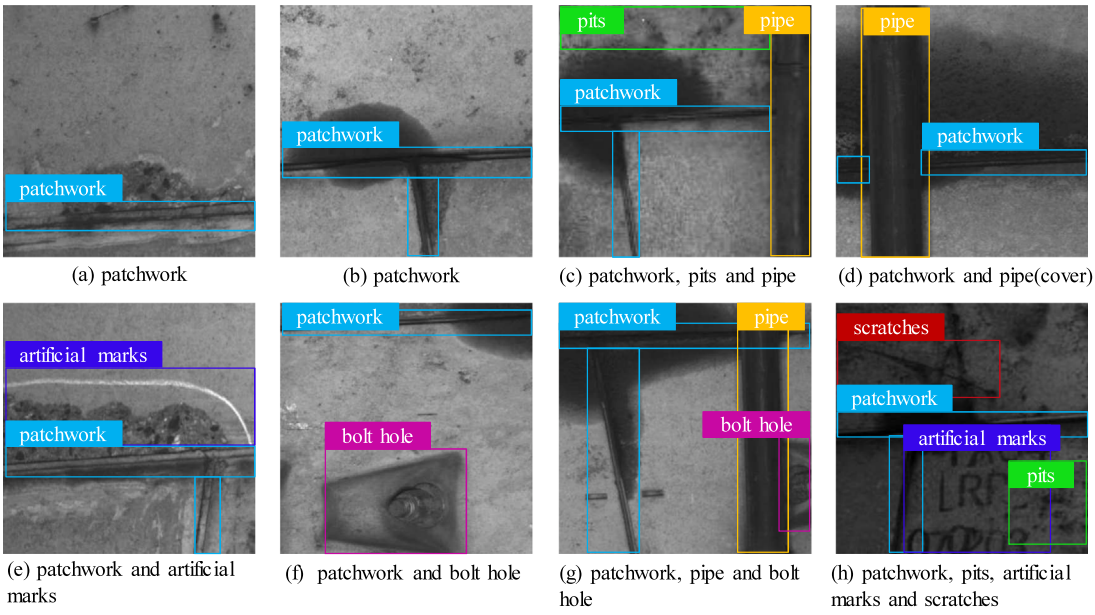
Specially, the contributions of our method are concluded as follows:

1. We introduce a deep learning-based framework based on Mask R-CNN to detect and segment leakages and spallings on the tunnel surfaces simultaneously.
2. We propose a path augmentation feature pyramid network (PAFPN) module to boost the information propagation of the backbone network of Mask R-CNN. The module is shared by object detection and mask prediction, leading to better performance of defect detection and segmentation.
3. We add a branch of edge detection to the end of the network to improve the accuracy of edge detection, and the result of edge detection is regarded as one of the network losses.
4. We design a simple yet effective algorithm to merge the detected defect regions belonging to the same one for subsequent tunnel quality assessment.

## 2. Characteristics of tunnel surface images

The tunnel surface images of our dataset present the following characteristics:

- Metro tunnels are underground tubular structures under the low light conditions. Although we have adopt lightings in the self-developed tunnel image acquisition equipment, some images are still dark.
- Leakages and spallings generally have a uncertain feature. The shape, size and extension direction of them are distinct.
- There are a large number of interferences on the tunnel surfaces, such as patchwork, pipes, pits, artificial marks, bolt holes and scratches.
- The gray value of the pipe and the leakage is very similar, both lower than that of the background.




(a) patchwork (b) patchwork (c) patchwork, pits and pipe (d) patchwork and pipe(cover)

(e) patchwork and artificial marks (f) patchwork and bolt hole (g) patchwork, pipe and bolt hole (h) patchwork, pits, artificial marks and scratches

**Fig. 2.** Some examples of leakage images containing various categories of interferences.

Fig. 2 shows some examples of tunnel surface images of our dataset. Fig. 2(a) is a spalling image containing patchwork. Fig. 2(b) is a leakage image containing patchwork. Fig. 2(c) contains patchwork, pits and pipe. Fig. 2(d) contains patchwork and pipe, in which a part of the leakage area is covered by the pipe. Fig. 2(e) contains patchwork and artificial mark. Fig. 2(f) contains patchwork and bolt hole. Fig. 2(g) contains patchwork, pipes and bolt hole. Fig. 2(h) is collected in dark environment, containing patchwork, pits, artificial marks and scratches.

It is difficult to recognize the defects on the tunnel surface images with these features. Defect detection algorithm based on partial features cannot effectively detect the leakages and spallings. To account for all of these features, we adopt a novel defect inspection method based on the Mask R-CNN.

## 3. Related work

In this section, we will discuss the related work related to R-CNN detectors and defect inspection respectively.

### 3.1. R-CNN detectors

Convolutional neural network (CNN) has a strong ability of image understanding, and can recognize the characteristics of objects. Regions with CNN (R-CNN) [34] is a two-step object detection framework, using a selective search approach to generate a number of candidate object regions. It was first introduced for object detection. Furthermore, many other researches have been proposed to enhance its performance. He et al. [35] proposed SPP-NET by applying spatial pyramid pooling (SPP) into R-CNN, which not only reduced the influence of input image size on the network, but also improved the precision of object detection. Girshick [36] further adopt region of interest pooling (RoIPool) into SPP-NET to propose an improved algorithm called Fast R-CNN, which greatly improved the speed of object detection. Ren et al. [13] proposed Faster R-CNN by introducing a novel region proposal network (RPN) to replace the previous slow selective search algorithm for candidate object regions generation. At present, Faster R-CNN has become the most popular algorithm in the field of object detection due to its prior performance. In order to extract object precisely from the images, object segmentation is demanded [37]. Long et al. [17] proposed the fully convolutional network (FCN) to detect the object regions by pixel level. It has been applied in many areas for object segmentation and has achieved remarkable results [24,38]. Mask R-CNN [20] is proposed for instance segmentation, combining Faster R-CNN framework and FCN algorithm to perform object detection and sematic segmentation task simultaneously.

### 3.2. Defect detection

In recent years, as the deep learning techniques prevail in computer vision, several studies have successfully applied to the automatic detection of defects in the field of structural health monitoring, such as steels [39,40], road pavements [41,42], bridges [43] and concrete buildings [44]. Feng et al. [45] proposed an automatic defect detection system based on artificial intelligence framework. A deep residual network (ResNet) was adopted as a classifier to realize defect detection and image classification for civil infrastructure. Fan et al. [41] developed a CNN-based multi-label classifier for pavement crack detection. Tabernik et al. [46] proposed a segmentation-based deep-learning architecture for the detection and segmentation of surface anomalies and was demonstrated on a specific domain of surface-crack detection. Chen and Jahanshahi [47] proposed a method to detect cracks in individual video frames by combining the convolutional neural network and Naive Bayes data fusion algorithm. These methods performed well in certain environments, but they are not qualified for tunnel defect detection due to the complicated characteristics of tunnel surface images. Makantasis et al. [48] employed a deep learning architecture to detect defects on the tunnel surfaces, and carried out the detection task by extracting high-level features through the convolution neural network (CNN). The experimental results show that CNN approach outperforms conventional pattern recognition algorithm, but the segmentation contour is relatively rough.

## 4. System overview

This section summarizes the whole process of the proposed framework for detecting tunnel surface defects. As shown in Fig. 3, the framework consists of three major parts: (1) tunnel surface image database construction; (2) defect detection and segmentation and (3) detected defects merging.

**Database Construction.** In this part, raw images are taken from real tunnels using a tunnel surface image acquisition system. Since our

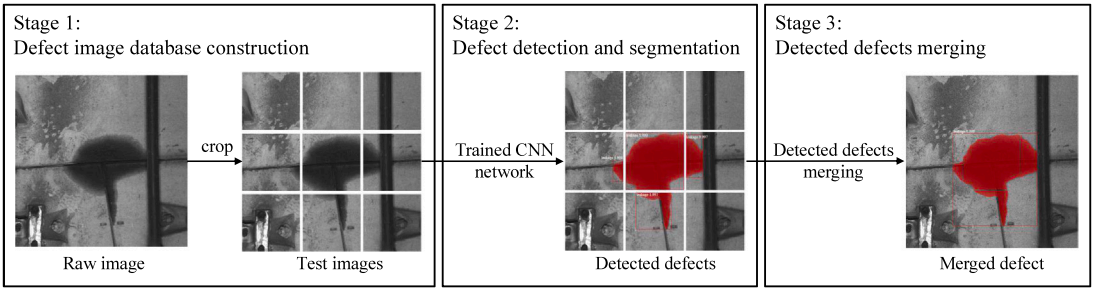



**Fig. 3.** Flowchart for detecting tunnel surface defects. Stage 1: Crop and manually annotate the raw images for the database construction; Stage 2: Send the test images to the trained CNN network for defect detection and segmentation; Stage 3: Merge the detected defect regions belonging to the same one.

original tunnel defect dataset is relatively small, we use data augmentation algorithm to increase the number of defect images. Then, the images are manually annotated by professional inspectors to establish tunnel surface image dataset for defect detection and segmentation model training and testing.

**Defect Detection and Segmentation.** In this section, we propose to design our framework based on Mask R-CNN. In order to fit the tunnel surface defect detection and segmentation task, we introduce a path augmentation feature pyramid network (PAFPN) and an edge detection branch into the original Mask R-CNN architecture. The proposed model can obtain enhanced features with both rich context information and edge information, leading to better performance of tunnel defect detection and segmentation results. In addition, considering the shape characteristics of leakages and spallings on the tunnel surfaces, we modify the aspect ratios of anchor boxes in the RPN network. Specific network design will be described in the later section.

**Defect Merging.** During the database construction process, a complete defect may be cropped into different images, as shown in Fig. 3 (Stage 1). In this section, we propose a method to merge the detected defect regions for subsequent engineering evaluation. First, we use the image location information previously stored during the image collection process to stitch the detected images containing leakage or spalling. Then, our proposed defect merge algorithm is used to search each connected defect region, so as to automatically merge the detected defect regions belonging to the same one.

## 5. Method

### 5.1. Database construction

In order to construct the tunnel defect database, we use a self-developed tunnel image acquisition equipment to collect tunnel surface images. As show in Fig. 4, the image acquisition system mainly consists of a running vehicle, CCD cameras, lightings, image collecting cards, a distance sensor and a computer. Due to the low light conditions in metro tunnels, it is necessary to adopt lightings to ensure the illumination for image collection. In order to carry out a comprehensive inspection of tunnel surface, we use multiple CCD cameras on the running vehicle to capture tunnel images. The location of each camera is calibrated to ensure that the overlapped image regions are fixed, which can avoid missing inspection. The configuration of these cameras is listed in Table 1. As the vehicle moves along the rails, the CCD cameras continuously capture images at a frequency appropriate to the vehicle's moving speed. Under the controlling of the computer, the system collects and stores tunnel surface images by the image collecting cards. In addition, in order to make the correspondence of the collected images with the real tunnel scene, we use the distance sensor to identify the location information of each collected image. The location information of the collected images is stored for offline defect detection.

After image collection, the raw images are cropped into the fixed resolution of 1500 × 1500. Images containing leakage and spalling are manually selected to build training set. Finally, 968 valid defect images are obtained. Then we use data augmentation algorithm to increase the number of defect images, since our original defect image dataset is relatively small, which is impossible to carry out robust training. Specifically, our dataset is augmented in three ways: random rotation, horizontal/vertical flipping and blurring. The rotation and flipping techniques can simulate different angles and different directions of the image collection by the cameras. Image blurring is used to simulate the captured images while the camera is shaking or failing to focus. Finally, the images containing defects are manually by the tool of LabelMe. In this study, all the images are divided into three parts: 80% of images are used for training, 10% of images are used for validating and 10% of images are used for testing.

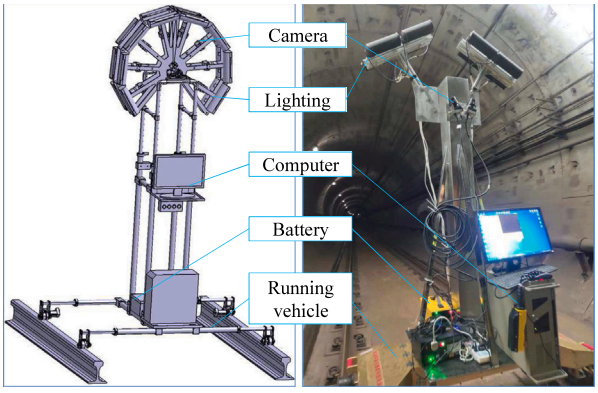



**Fig. 4.** The equipment for tunnel surface image data collection. (a) is the self-designed model; (b) is the image acquisition equipment.

**Table 1**
The configuration of CCD cameras.

| Resolution | Pixel size | Maximum frame rate | Interface |
|---|---|---|---|
| 12288 × 1 | 7μm×7μm | 66k Hz | CameraLink 3.0 |

### 5.2. Defect detection and segmentation

Our proposed tunnel surface defect detection and segmentation model is based on Mask R-CNN, which is an extension of Faster R-CNN, adding a parallel branch for predicting object mask to the existing branch for object classification and bounding box location. Therefore, Mask R-CNN is also composed of two stages. In the first stage, hierarchical features are extracted using a ResNet-FPN backbone network. Then the candidate object bounding boxes are generated through a Region Proposal Network (RPN). In the second stage, in addition to predict the class and bounding box recognition, Mask R-CNN also generates a binary mask for each RoI by using a fully convolutional network (FCN).

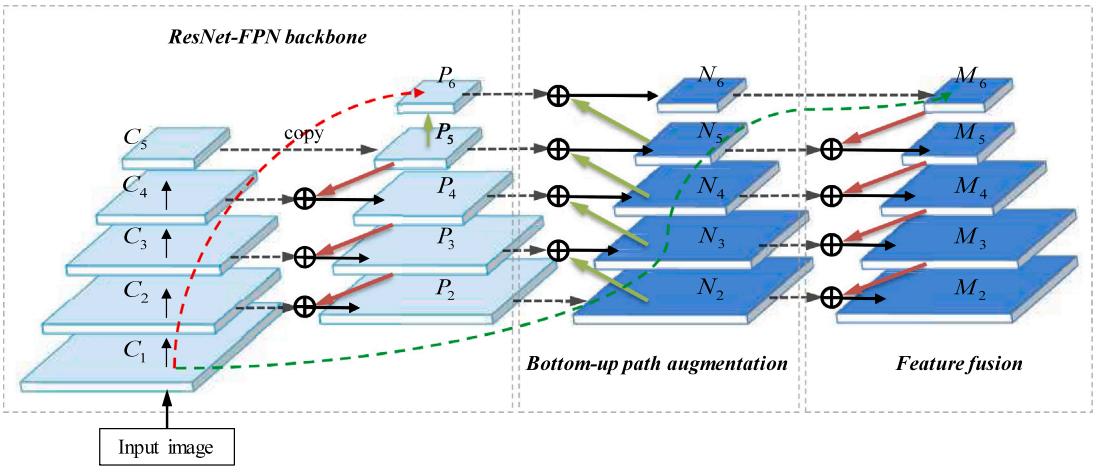



**Fig. 5.** An illustration of the proposed PAFPN module. Dotted gray lines represent the copy operation; Solid red lines represent the bilinearly up-sampling operation; Solid green lines represent the convolutional down-sampling operation. Dotted red line shows the original long path from low layers to the topmost one; Dotted green line shows the new "short" path from low layers to the topmost one. Bottom-up path augmentation is used to construct enhanced features; Feature fusion layers are used to merge context information to lower-level layers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Mask R-CNN uses RoI alignment (RoIAlign) to replace RoI pooling operation of Faster R-CNN, which performs coarse quantizations that lead to misalignments of RoI and the extracted features.

*5.2.1. Path augmentation feature pyramid network*

In CNN detectors, high-level features are strongly related to entire objects, while low-level features are more related to local information of the objects. Therefore, it is necessary to introduce a top-down path in FPN to propagate high-level features to enhance all features for object classification. However, this single propagation approach is not fit for object detection, because there is a long path passing through even 100+ layers from low layers to the topmost one, resulting in the loss of localization information because of pooling and deconvolution during feature extraction process. Inspired by this observation, we take FPN as a baseline and significantly enhance it. Specially, we adopt a path augmentation feature pyramid network (PAFPN) to enhance the feature hierarchy with rich low-level features by adding a bottom-up path augmentation module and a feature fusion operation module. In this way, we introduce a "short" path consisting of less than 10 layers from low layers to the topmost one.

Fig. 5 shows the proposed PAFPN module in details. Each cube represents a corresponding feature tensor. Dotted gray lines represent the copy operation. Solid red lines represent the bilinearly up-sampling operation. Solid green lines represent the convolutional down-sampling operation. Dotted red line shows the original long path from low layers to the topmost one. Dotted green line shows the new "short" path from low layers to the topmost one. In the original ResNet-FPN backbone network, features are extracted from the final convolutional layer of conv1–conv5 parts of ResNet101, which are called $C_1$, $C_2$, $C_3$, $C_4$ and $C_5$ in this paper. Based on the bottom-up network architecture, the feature extraction layers compute hierarchical feature maps. Because of the maxpooling operations between these convolutional layers, the feature maps generated by each layer are in the size of 1/2, 1/4, 1/8, 1/16 and 1/32 of the input image respectively. Feature maps generated by FPN are represented by $P_2$, $P_3$, $P_4$, $P_5$, and $P_6$. From $P_6$ to $P_2$, the spatial size of feature maps is gradually up-sampled with factor 2. Concretely, $P_i$ is up-sampled up to two times larger and concatenated with $C_{i-1}$ to generate $P_{i-1}$. Note that $P_6$ is directly down-sampled with factor 2 from $P_5$.

We follow FPN to define the feature maps with the same spatial size generated in PAFPN. The feature maps of the added bottom-up path augmentation module are represented as $N_2$, $N_3$, $N_4$, $N_5$ and $N_6$ corresponding to $P_2$ to $P_6$. The concrete operations for bottom-up path augmentation module are illustrated in Fig. 6. Firstly, $N_i$ is processed by a $3 \times 3$ convolutional layer with stride 2 to down-sampled to two times. Then the down-sampled feature map is concatenated with $P_{i+1}$. At last, the fused feature map goes through another $3 \times 3$ convolutional layer to generate new feature map $N_{i+1}$. Note that $N_2$ is simply $P_2$, without any processing. Then, feature fusion operations are carried out to incorporate higher level feature maps to the lower level ones for contextual feature fusion. We use $M_2$, $M_3$, $M_4$, $M_5$, $M_6$ to denote feature maps generated in new up-bottom feature fusion path. The concrete operations for feature fusion are illustrated in Fig. 7. $M_i$ is bilinearly up-sampled up to two times larger. Then the generated larger feature map is concatenated with $N_{i-1}$ to generate $M_{i-1}$.

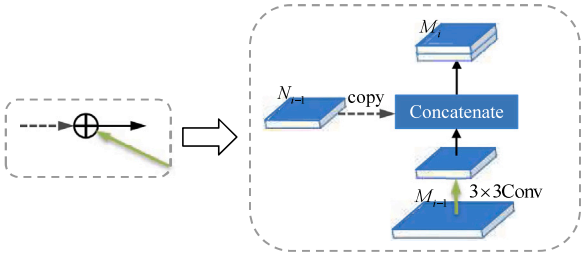



**Fig. 6.** Bottom-up path augmentation. $N_i$ is processed by a $3 \times 3$ convolutional layer with stride 2 to down-sampled to two times. Then the down-sampled feature map is concatenated with $P_{i+1}$. Finally, the fused feature map goes through another $3 \times 3$ convolutional layer to generate new feature map $N_{i+1}$.

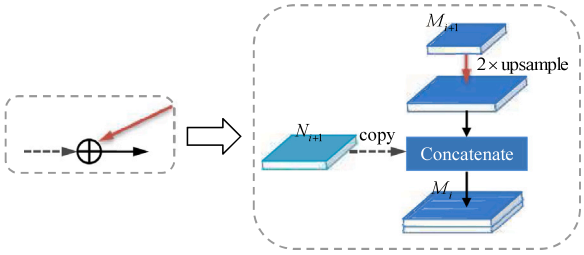



**Fig. 7.** Feature merging operations. $M_i$ is bilinearly up-sampled up to two times larger. Then the generated larger feature map is concatenated with $N_{i-1}$ to generate $M_{i-1}$.

Meanwhile, due to the bounding boxes surrounding the leakage and spalling typically shaped as elongated rectangles, we change the aspect ratios of anchor boxes in the RPN network to 1:1, 1:2, 2:1, 1:4 and 4:1. The changed anchor boxes are illustrated in Fig. 8.

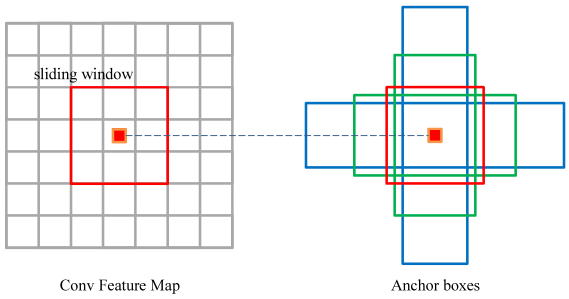



**Fig. 8.** Anchor boxes generation. At each sliding-window location, 5 aspect ratios (1:1, 1:2, 2:1, 1:4 and 4:1) are applied.

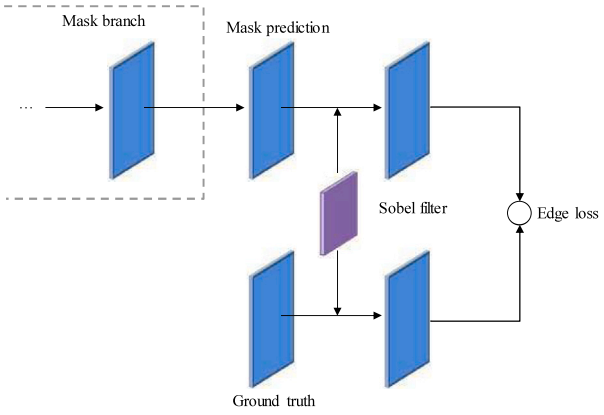



**Fig. 9.** An illustration of the edge detection branch. We use Sobel edge detection filter to perform convolution operation with the mask prediction results of Mask R-CNN and the corresponding ground truth of tunnel defect areas respectively. The difference between the edge detection results is calculated as the edge loss and added into the network losses.

#### 5.2.2. Edge detection branch

In the defect inspection task, high response to defect edge is an important factor for segmenting defects accurately. With this in mind, we propose a branch of edge detection to the end of the Mask R-CNN to enable the network focus on the edge information of defects. In this way, the modified network can achieve higher accuracy of the edges of the segmentation results.

Fig. 9 shows the structure of the proposed edge detection network. The inputs of the network are the prediction outputs of the mask branch of Mask R-CNN and the corresponding ground truth of tunnel defect areas. To obtain the edge detection results, we use the Sobel edge detection filter to perform convolution operation with the inputs respectively. Then, the difference between the edge detection results is calculated as the edge loss and is added into the network losses.

In this paper, the edge detection loss $L_{edge}$ is defined as the root mean squared error (RMSE) between predicted results and true ones:

$$L_{edge} = \frac{1}{N} \sum_{1 \le i \le N} (|y_i - \hat{y}_i|)^2, \tag{1}$$

where $y_i$ denotes the edge detection result of the prediction output for the $i$th image, $\hat{y}_i$ denotes the edge detection result of the corresponding ground truth for the $i$th image. $N$ denotes the number of images.

The structure of the proposed tunnel surface defect detection and segmentation model is illustrated in Fig. 10, which includes four major parts: (1) a backbone network combined with proposed PAFPN to obtain multi-scale features with rich defect localization information; (2) a region proposal network for generating region proposals which may contain defects; (3) a prediction module for object classification, bounding box regression and object mask prediction and (4) an edge detection branch to calculate edge detection loss $L_{edge}$.

### 5.3. Loss function

Except for edge detection loss, the loss for classification, box regression and the mask generation are the same as those of Mask R-CNN. To each sampled RoI, a multi-task loss $L$ is applied for the joint training of the four tasks:

$$L = L_{class} + L_{box} + L_{mask} + L_{edge}, \tag{2}$$

where $L_{class}$ is the loss for classification, $L_{box}$ is the loss for bounding box regression, $L_{mask}$ is the loss for mask prediction and $L_{edge}$ is the loss for edge prediction. $L_{class}$ and $L_{box}$ are defined the same as in Faster R-CNN [13]:

$$\begin{aligned} L_{class} + L_{box} &= \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ &+ \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i - t_i^*), \end{aligned} \tag{3}$$

where $i$ denotes the index of an anchor in a mini-batch. $p_i$ denotes the predicted probability of anchor $i$ being an object. $p_i^*$ denotes the ground truth label (binary) of anchor $i$. $p_i^* = 1$ if anchor $i$ is positive, and $p_i^* = 0$ if anchor $i$ is negative. $t_i$ is a vector denoting the predicted 4 parameterized coordinates. $t_i^*$ denotes the ground truth coordinates. $N_{cls}$ and $N_{reg}$ are used to normalize $clc$ and $reg$ terms, and $\lambda$ is used to balance the two terms.

Log loss is used for the classification loss over object and non-object:

$$L_{cls}(\{p_i, p_i^*\}) = -p_i^* \log p_i^* - (1 - p_i^*) \log(1 - p_i^*), \tag{4}$$

we use smooth $L1$ [36] for the box regression loss:

$$L_{reg}(\{t_i, t_i^*\}) = L_1^{smooth}(t_i - t_i^*), \tag{5}$$

The mask prediction loss is the average binary cross-entropy loss:

$$L_{mask} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} y_{ij} \log \circ y_{ij}^k + (1 - y_{ij}) \log (1 - \circ y_{ij}^k), \tag{6}$$

### 5.4. Defect merging

Considering that a complete defect region may be cropped into different images during the database construction process. Therefore, we propose a method to merge the detected defect regions for subsequent engineering evaluation.

Firstly, we stitch each detected image containing leakages or spallings using the image location information previously stored during the image collection process. Suppose that the detected images are denoted by $X = \{X_1, \ldots, X_N\}$. $K_i$ represents the number of detected defect regions in the $i$th image. $T_k = \{T_i^k, i = 1, \ldots, N; k = 1, \ldots, K_i\}$, where $T_i^k$ represents the $k$th detected defect region in the $i$th image. $S_k = \{S_i^k, i = 1, \ldots, N; k = 1, \ldots, K_i\}$, where $S_i^k$ represents the mask value of the defect region $T_i^k$. Let $L_i$ represents the number of neighbors of the $i$th image. $B_l = \{B_i^l, i = 1, \ldots, N; l = 1, \ldots, L_i\}$, where $B_i^l$ represents the index of the $l$th neighbor of $i$th image. Then, we calculate the minimum distance $d$ between each defect region and its adjacent defect region. As shown in Fig. 11, the contour of defect region is represented as a set of points $\{P_t\}$. The minimum distance $d$ between two different defect regions is defined as follows

$$C(P_t, P_j) = \sqrt{(P_{tx} - P_{jx})^2 + (P_{ty} - P_{jy})^2}, \tag{7}$$

$$d = min(min(C)), \tag{8}$$

where $C$ is a matrix of the distances between point set $\{P_t\}$ and $\{P_j\}$; $P_{tx}$, $P_{ty}$, $P_{jx}$ and $P_{jy}$ denote the coordinates of points in $\{P_t\}$ and $\{P_j\}$, respectively.

For each defect region and its adjacent defect regions, the algorithm merges them when they meet the following two conditions. One is that the minimum distance $d$ between them is less than the given threshold

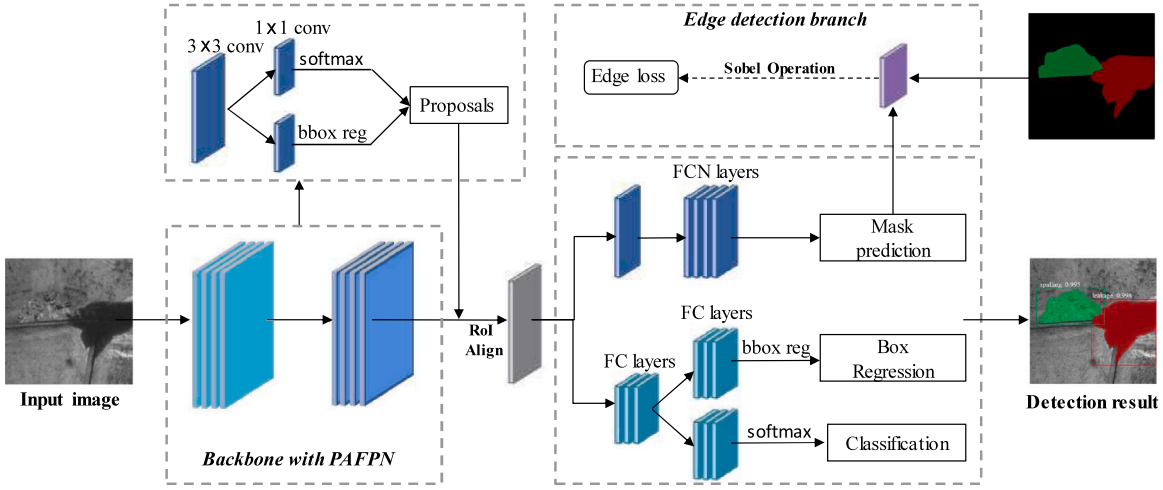



**Fig. 10.** An illustration of the architecture of the proposed defect detection and segmentation network, which consists of four parts. Backbone network with PAFPN is to generate multi-scale features with rich defect localization information; RPN is to generate region proposals which may contain defects; Prediction module includes object mask prediction, object classification and bounding box recognition.; Edge detection branch is to calculate edge detection loss $L_{edge}$.

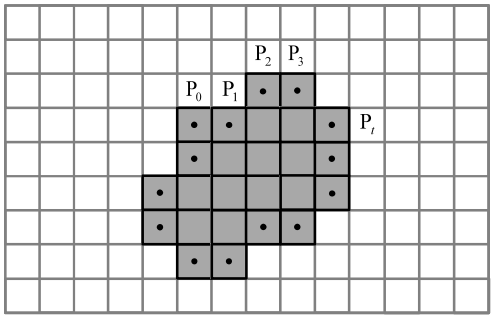



**Fig. 11.** The contour of defect region is represented as a set of points $\{P_i\}$.

**Table 2**
Sample number for each defect type.

| Type | Total | Training | Validation | Testing |
|---|---|---|---|---|
| Normal | 5136 | 3200 | 968 | 968 |
| Leakage | 3920 | 3136 | 392 | 392 |
| Spalling | 4580 | 3664 | 458 | 458 |
| Leakage-Spalling | 1180 | 944 | 118 | 118 |

**Algorithm 1** The pseudo code of detected defects merging

**Input:** $X$ : dataset of detected images;
**Output:** $T$ : dataset of detected defects of $X$;
  **Initialize:** $N$: the cardinality of $X$;
  **while** $i = 1$ to $N$ **do**
    Extract $T = \{T_i^1, ..., T_i^{K_i}\}$ from $X_i$;
    Extract $B = \{B_i^l, i = 1, ..., N; l = 1, ..., L_i\}$ of $X_i$;
  **while** $i = 1$ to $N$ **do**
    **while** $l = 1$ to $L_i$ **do**
      $index = B_i^l$;
      **while** $k = 1$ to $K_i$ **do**
        **while** $m = 1$ to $K_{index}$ **do**
          Calculate $d$ between $T_i^k$ and $T_{index}^m$;
          **if** $d \leq$ **Dis then**
            **if** $S_i^k = S_{index}^m$ **then**
              Merge $T_i^k$ and $T_{index}^m$;
              Draw the bounding box of the new merged defect region;
              Update $T$;

represented by $Dis$. The other is that the defect regions belonging to the same defect category, which is determined by comparing their mask values. Through this algorithm, each connected defect region is found by investigating each neighbor of the active defect regions for regional membership. The pseudo code for detected defects merging process is given in algorithm 1.

## 6. Experiments and results

The above analysis of the proposed method provides the feasibility for automatic defect detection and segmentation for metro tunnel surfaces. In this section, the accuracy of our tunnel surface defect detection method is verified.

### 6.1. Dataset

The dataset applied in this study is collected from a series of metro lines in China. In order to avoid overfitting, the data augmentation algorithm is used to increase the number of images containing defects. After data augmentation, we obtain 9680 images containing defects with the resolution of 1500 × 1500. To build the dataset, the defect images are annotated manually. Specifically, the final dataset contains four types of tunnel surface images, including normal, leakage-only, spalling-only and leakage-spalling images, as shown in Fig. 12. The sample number of each image type used in each stage of the experiment is listed in Table 2.

### 6.2. Training process

In this study, given a tunnel defect image with corresponding ground truth, we adopt the IOU (Intersection over Union) strategy [13] to define the positive and negative samples of anchors. We define the anchors with IOU score greater than 0.7 as positive, and those with IOU score less than 0.3 are defined as negative, so that most of region proposals can be discarded. Then, in each mini-batch iteration, all the anchors selected are used to calculate the classification and coordinate regression loss. We train about 50,000 iterations in total with weight decay of 0.0001 and momentum of 0.9 respectively. The learning rate is set to 0.001 initially. After 30,000 iterations, we set the learning rate to 0.0001. The experimental environment is described as follows: Deep learning open source framework Tensorflow 2.0, Windows 10, Python

**Table 3**
Comparison of mAP results for three types of images using different methods.

| Methods | +PAFPN | +Edge Detection | AP | | mAP |
|---|---|---|---|---|---|
| | | | Leakage-only | Spalling-only | Leakage-Spalling |
| Mask R-CNN | | | 74.35% | 82.62% | 77.37% |
| Method-A | ✓ | | 83.30% | 91.12% | 83.62% |
| Method-B | | ✓ | 78.79% | 84.29% | 85.58% |
| Our method | ✓ | ✓ | **85.35%** | **93.68%** | **90.57%** |

**Table 4**
Comparison of error rate results for three types of images using different methods.

| Methods | + PAFPN | + Edge Detection | Leakage-only | Spalling-only | Leakage-Spalling |
|---|---|---|---|---|---|
| Mask R-CNN | | | 2.88% | 1.50% | 2.23% |
| Method-A | ✓ | | 0.95% | 0.52% | 0.72% |
| Method-B | | ✓ | 2.03% | 1.31% | 1.73% |
| Our method | ✓ | ✓ | **0.61%** | **0.42%** | **0.57%** |

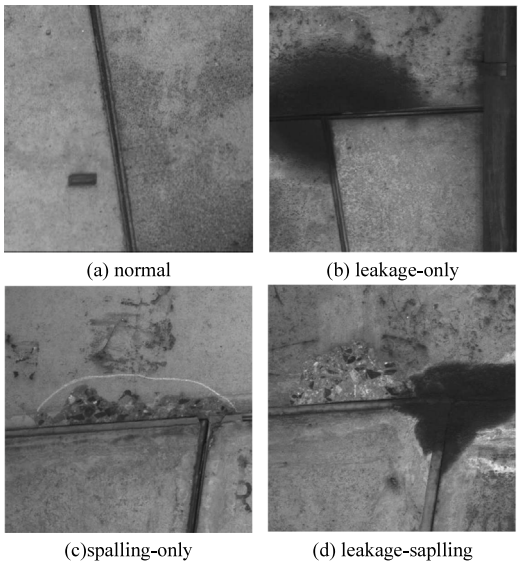



**Fig. 12.** Four typical images used in this study: (a) normal; (b) leakage-only; (c) spalling-only; (d) leakage-spalling.

2.7, Intel Core I7-8700, and GTX 1080 graphical processing unit (GPU) with 8-GB memory.

With the limited training set, we apply the transfer learning technique [49] for the pre-training process. Specially, we use the COCO dataset [24] to pre-train the network, which is containing 328k images with more than 2.5 million labeled instances. After that, the weights are transferred to initialize the defect detection network. In the final training process, we use the established tunnel surface dataset to fine-tune the layers of the pertained detection model.

### 6.3. Metrics

For defect detection task, mean Average Precision (mAP) is usually employed to quantify the performance of the algorithm. We assume that there are $K$ classes to be detected, mAP is calculated using the well-known metrics (precision and recall) as follows:

$$precision = \frac{TP}{TP + FP}, \tag{9}$$

$$recall = \frac{TP}{TP + FN}, \tag{10}$$

$$AP = \int_0^1 P(R)\, dR, \tag{11}$$

$$mAP = \frac{1}{K} \sum_{i=1}^{K} AP_i, \tag{12}$$

where TP (true positive) denotes the number of tunnel surface defects, which are classified correctly; FP (false positive) denotes the number of the tunnel surface defects which are predicted as normal ones; and FN (false negative) denotes the number of normal surfaces, which are identified as defective ones.

In addition, we use the error rate [38] as another indicator to evaluate the segmentation performance of our proposed method, which is defined as the ratio between the mislabeled pixels and the total pixels

$$error\ rate = \frac{N_{false}}{N_{tatal}}, \tag{13}$$

where $N_{false}$ denotes the number of mislabeled pixels, $N_{total}$ denotes the number of total pixels in the images.

### 6.4. Ablation study for proposed modules

Compared with the original Mask R-CNN architecture, there are two main changes in our proposed defect detection and segmentation method. One is that we apply a PAFPN module to the original Mask R-CNN to obtain feature maps with rich low-level information. The other one is that we propose an edge detection branch and add it to the end of the network to improve the accuracy of edge segmentation results. In order to test the influence of the added PAFPN module and the edge detection branch, we establish three different network frameworks based on Mask R-CNN, which are represented by Method-A, Method-B and our method respectively. Method-A stands for the Mask R-CNN network with the PAFPN; Method-B stands for the Mask R-CNN network with the edge detection branch. Our method combines both two improvements. In this section, three group experiments are implemented for detection and segmentation of leakage-only, spalling-only and leakage-spalling images.

Fig. 13 shows some examples of defect detection results using different methods. The leakage areas are outlined by red boxes. The green boxes represent recognized spalling areas. It can be observed that our method has better performance in both defect detection and segmentation of tunnel surface compared with original Mask R-CNN. For example, in the tested results of leakage-only images, Mask R-CNN loses some edge information and mistakenly detects the background as the leakage areas (line 1), while the defect detection results of our method are closer to the ground truth. This is because the proposed PAFPN module enables the backbone network to obtain features with rich low-level information, making the RPN generate more accurate candidate boxes. In addition, the segmentation contour of our method is more distinct while the segmentation contour of Mask R-CNN is relatively rough. As can be seen from the tested results of leakage-spalling images (line 3), the adjacent leakage and spalling areas detected by Mask R-CNN are severely overlapped and the boundary of the two defects cannot be distinguished. While Method-B obtains more clear

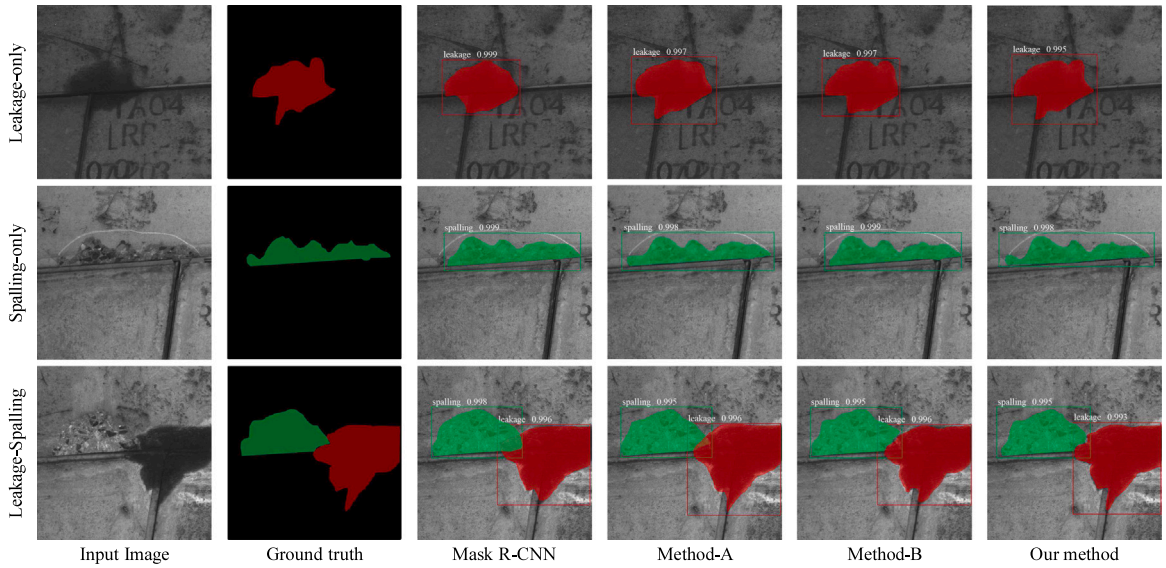



**Fig. 13.** Examples of the defect detection results using different methods. The leakage areas are outlined by red boxes. The green boxes represent recognized spalling areas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

boundary of the adjacent defect areas, which proves the effectiveness of the proposed edge detection branch.

Tables 3 and 4 show the mAP and error rate results for three types of defect images respectively. Compared with other methods, our method achieves the best performance in AP score and error rate results based on all three types of tunnel defect images, which confirms the effectiveness of our method in defect identification on tunnel surfaces. Fig. 14 shows the bar charts of the changes in mAP and error rate of Method-A, Method-B and our method compared with the original Mask R-CNN. From Fig. 14, we can observe that the AP score and the error rate of test results vary significantly with the proposed PAFPN algorithm for the leakage-only images. The AP score of Method-A is significantly improved by 8.95% (from 74.35% to 83.30%). Meanwhile, the error rate of Method-A is significantly reduced by 1.93% (from 2.88% to 0.95%). For spalling images, the AP score of Method-A is also significantly improved by 8.5%. The results prove that the PAFPN algorithm can successfully avoid information loss of low-level features and enhance the feature extraction ability of the original backbone network of Mask R-CNN. As for edge detection branch, the AP score of leakage-only images is significantly increased by 4.44% using Method-B, and the error rate is reduced by 0.85%. While the AP score for spalling detection is only improved by 1.67%. The result demonstrates that the edge information has more influence on leakage detection than spalling detection. This is because the feature of spalling is relatively simple, whereas the shape of leakage is usually complex. Therefore, the added edge detection branch is more effective for leakage detection task.

Through the above defect detection experiments, it is proved that our method successfully improves the accuracy for leakage and spalling detection. Our proposed PAFPN algorithm could better capture features of the tunnel defects, and the added edge detection branch could improve the accuracy of the edges of tunnel defect segmentation results.

### 6.5. Comparative experiment

In order to evaluate the effectiveness of our proposed method for detecting and segmenting tunnel defects, we compare our method with both traditional methods and state-of-the-art learning-based methods. For a fair comparison, the methods for comparison are all performed on our tunnel surface image database.

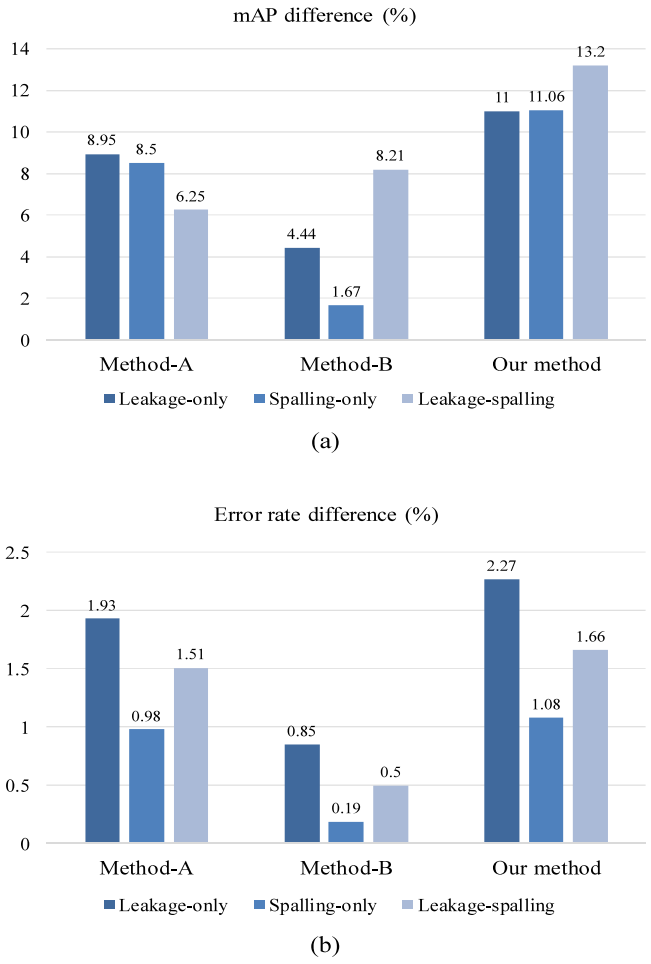



**Fig. 14.** Bar charts of the changes in mAP and error rate by Method-A, Method-B and our method compared with the original Mask R-CNN. (a) shows the changes in mAP by different methods compared with the original Mask R-CNN; (b) shows the changes in error rate by different methods compared with the original Mask R-CNN.

*6.5.1. Comparing against the traditional methods*

As shown in Fig. 2, there are a number of complex interferences on the tunnel surfaces, such as patchwork, pipes and pits, et al. These interferences make it difficult to recognize the defects from tunnel surfaces, especially for leakage identification. Traditional segmentation algorithms do not require any training data. If they can achieve competitive defect detection performance over the CNN detector, it is better to apply them instead of using algorithms based on deep learning techniques. In order to prove that the proposed method can accurately segment tunnel defects on tunnel surfaces, we first compare our method with traditional segmentation methods, including Otsu Algorithm (OA) [50], Watershed Algorithm (WA) [51] and Region Growing Algorithm (RGA) [52]. Fig. 15 shows some examples of leakage segmentation results using these different methods, among which the black areas are the leakage areas detected by OA, WA and RGA, and the red areas are the leakage areas detected by our method. The testing results of segmentation error rate are listed in Table 5.

From Fig. 15, we can observe that our method outperforms other methods for all types of images containing various interferences. For the defect images containing patchwork (line 1), the three traditional methods identify the patchwork as defect areas by mistake. Among them, RGA is better than OA and WA in restraining the interference caused by patchwork. For the defect images containing patchwork and pipes (line 2, line 3 and line 5), due to the colors of pipes are similar to the leakage areas, the three traditional methods mistakenly identify the pipes as leakage areas, resulting in the detected leakage areas are larger than the actual areas. Especially for OA and WA, they are equally affected by pipes. Our method is least affected by pipes and performs well than other methods. For the interference called bolt holes (line 4 and line 5), our method can identify it correctly while the traditional methods cannot distinguish them from gray leakage areas. For the defect images containing pits or scratches (line 6), our method and RGA are not affected by these background noises and perform well than OA and WA. OA is sensitive to these interferences and cannot remove them from the image. WA is affected most, seriously missing the information of the defect areas.

Compared with other methods, our proposed method gets the best performance overall. The tunnel defect detection and segmentation method proposed in this paper is effective in restraining these interferences. Traditional methods only rely on low-level features, such as gray value, edges, textures, etc. These low-level features are easily to be disturbed by the background noises on tunnel surfaces, which leads to the high detection error rate.

In this experiment, 100 representative images with various interferences are collected to test the average error rate of the six different methods. As shown in Fig. 16, compared with these traditional methods, our method significantly reduces the segmentation error rate, which shows the advantage of deep learning techniques. The average error rate of OA, WA, and RGA is 36.59%, 39.68%, 23.97% respectively, while the average error rate of our method is only 0.64%.

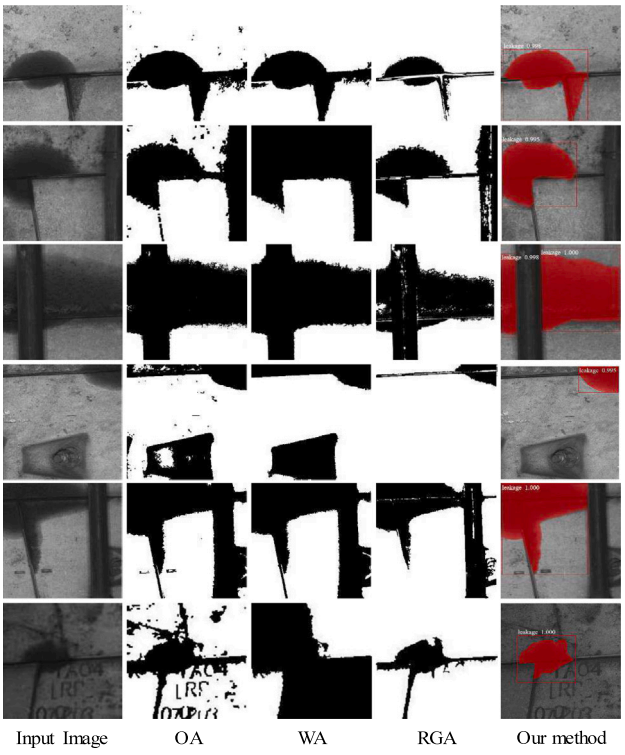



**Fig. 15.** Examples of the leakage detection results using traditional methods and our method, among which the black areas are the leakage areas detected by OA, WA and RGA, and the red areas are the leakage areas detected by our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

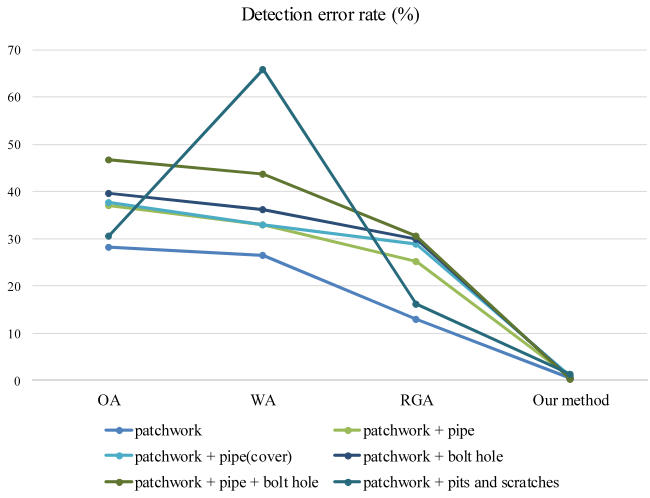



**Fig. 16.** Detection error rate of traditional methods and our method for leakage recognition.

*6.5.2. Comparing against the state-of-the-art methods*

For further validation of the proposed tunnel defect detection and segmentation method, we compare our method with state-of-the-art instance segmentation methods, including MS R-CNN [53], SpineNet [21], ResNeSt [23] and CenterMask [22]. Table 6 shows the performance comparisons of different methods in tunnel defect detection and segmentation. The performance of these comparison methods is measured using their open-sourced implementation and the FLOPs and params are obtained from their papers.

We first report the experimental results of MS R-CNN, CenterMask and our method using the same baseline ResNet. From Table 6, we can observed that CenterMask's performance is on par with MS R-CNN but with much fewer FLOPs. Under the same ResNet101 backbone, our method significantly improves the mAP score by 9.18% (from 79.58% to 88.76%) compared to CenterMask. Then we report the experimental results of SpineNet, ResNeSt and our method under the same baseline Mask R-CNN. It can be observed that our method requires extra computation and memory but outperforms SpineNet and ResNeSt in terms of both box mAP sore and mask error rate. The results demonstrate the defect detection and segmentation accuracy improvement of our method by adding PAFPN and edge detection branch.

Fig. 17 shows the comparison defect detection results using these different methods. The leakage areas are outlined by red boxes. The green boxes represent recognized spalling areas. It can be observed

**Table 5**
Comparison of error rate results for leakage recognition using traditional methods and our method.

| Category | patchwork | pipe | cover | bolt hole | pits and scratches | OA [50] | WA [51] | RGA [52] | Our method |
|---|---|---|---|---|---|---|---|---|---|
| (1) | ✓ | | | | | 28.19% | 26.52% | 12.97% | **0.52%** |
| (2) | ✓ | ✓ | | | | 36.97% | 32.85% | 25.20% | **0.75%** |
| (3) | ✓ | | ✓ | | | 37.58% | 32.93% | 28.82% | **0.83%** |
| (4) | ✓ | | | ✓ | | 39.53% | 36.26% | 29.95% | **0.36%** |
| (5) | ✓ | ✓ | | ✓ | | 46.78% | 43.72% | 30.66% | **0.18%** |
| (6) | ✓ | | | | ✓ | 30.54% | 65.83% | 16.27% | **1.23%** |
| Average | | | | | | 36.59% | 39.68% | 23.97% | **0.64%** |

**Table 6**
Tunnel defect detection and segmentation results using different instance segmentation methods.

| Methods | Backbone | #FLOPs | #Params | box mAP | error rate |
|---|---|---|---|---|---|
| MS R-CNN | ResNet50 FPN [53] | 852.2B | 48.0M | 78.17% | 2.82% |
| | ResNet101 FPN [53] | 1011.5B | 69.8M | 79.78% | 2.23% |
| Mask R-CNN | SpineNet96 [21] | 446.3B | 57.5M | 81.38% | 2.66% |
| | SpineNet143 [21] | 712.5B | 79.1M | 82.78% | 2.09% |
| Mask R-CNN | ResNeSt50 [23] | 150.3B | 25.6M | 81.09% | 2.59% |
| | ResNeSt101 [23] | 292.8B | 48.0M | 83.23% | 1.98% |
| CenterMask | ResNet50 FPN [22] | 464.9B | 51.2M | 78.09% | 2.79% |
| | ResNet101 FPN [22] | 631.4B | 70.1M | 79.58% | 2.25% |
| Our method | ResNet50 PAFPN | 894.5B | 50.3M | 85.19% | 0.62% |
| | ResNet101 PAFPN | **1033.2B** | **71.5M** | **88.76%** | **0.51%** |

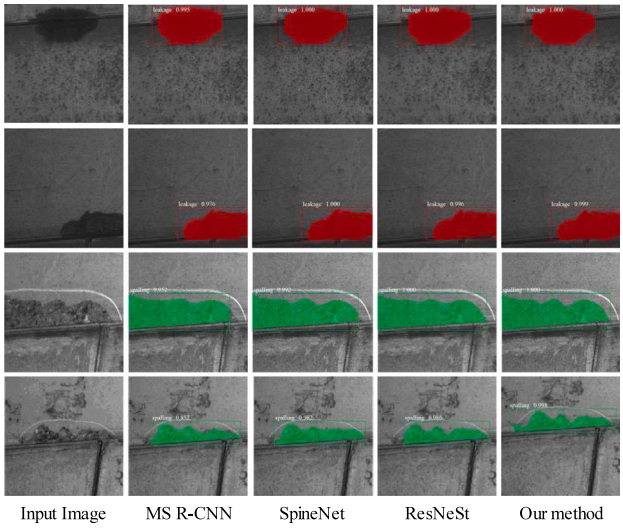



**Fig. 17.** Examples of the defect detection results using state-of-the-art methods and our method. The leakage areas are outlined by red boxes. The green boxes represent recognized spalling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that our well-trained model tends to be less likely to propose bounding boxes which loss some information of defects. This indicates that the features extracted by our method can make the RPN generate more accurate candidate boxes. Moreover, for the defects with complex shapes, the segmentation contour of our method is more sharp and fine while that of other methods is relatively rough. This indicates that our proposed edge detection branch enables the network to focus on the edge information of defects.

### 6.6. Defect merging

To obtain the appropriate distance as the threshold for adjacent defect regions merging, we select 10 thresholds from 5 pixel to 50 pixel for comparison by step size 5 pixel. In this experiment, we randomly collect 100 adjacent images in the database and manually count that

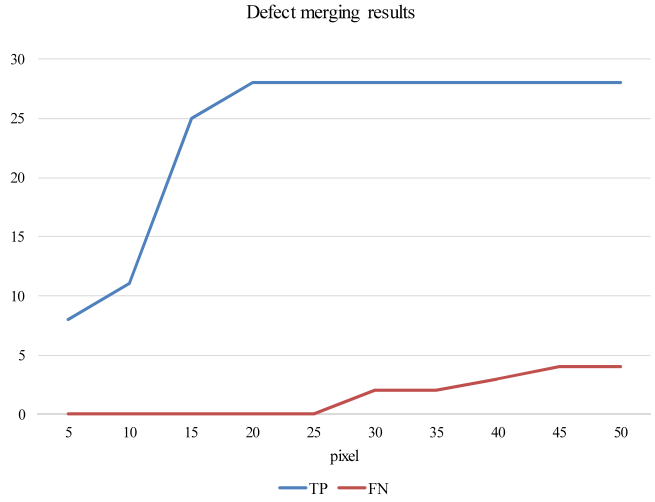



**Fig. 18.** The TP and FN results of the 10 different thresholds for detected defect areas merging.

there are a total of 28 defect areas to be combined. We employ TP and FN to test the performance of the 10 different thresholds, where TP (true positive) denotes the number of detected defect areas which are merged correctly; FN (false negative) denotes the number of detected defect areas which are merged mistakenly. Table 7 and Fig. 18 show the test results of the 10 different thresholds. It can be observed that the accuracy of defect merging increases with the increase of threshold value and reaches the highest value when the threshold is set to 25 pixel. However, when the threshold value exceeds 30 pixels, the number of false merged areas begins to increase as threshold value increases. Therefore, we set the threshold $Dis$ to 25 pixels to obtain the best performance.

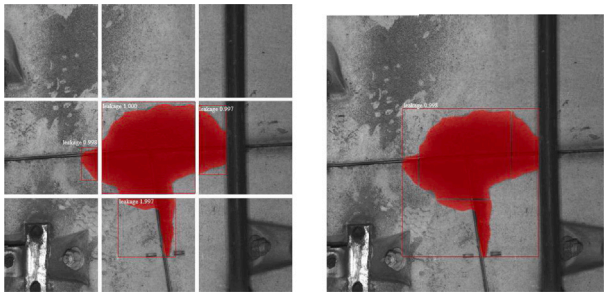
Fig. 19 shows an example of the detected defects after the process of defect merging. Fig. 19(a) shows the defect detection and segmentation results, where the regions outlined by red boxes represent the detected leakage regions. It can be observed that the entire defect region is segmented into different small images. Then, according to Section 3-D, the algorithm merges the detected defect regions belonging to the same one. The connected defect region is found by investigating each neighbor of every active region for regional membership. Fig. 19(b) shows the result of defect merging process. We can observe that the algorithm performs well. Owing to the prior knowledge about the process of database construction and defect detection and segmentation, the defect merging algorithm achieves reliable performance.

## 7. Conclusion

In this paper, a novel tunnel surface defect detection and segmentation method based on the Mask R-CNN is presented. The detection of leakage and spalling is necessary for tunnel surface inspection and maintenance. By applying the PAFPN module and the edge detection branch into the original Mask R-CNN framework, our well-trained model can automatically detect and segment leakage and spalling in the

**Table 7**
The TP and FN results of the 10 different thresholds for detected defect areas merging.

| Metrics | 5 pixel | 10 pixel | 15 pixel | 20 pixel | 25 pixel | 30 pixel | 35 pixel | 40 pixel | 45 pixel | 50 pixel |
|---|---|---|---|---|---|---|---|---|---|---|
| TP | 8 | 11 | 25 | 28 | **28** | 28 | 28 | 28 | 28 | 28 |
| FN | 0 | 0 | 0 | 0 | **0** | 2 | 2 | 3 | 4 | 4 |



(a) detected defect images (b) defect merging result

**Fig. 19.** An example of defect merging: (a) detected defect images; (b) result of defect merging process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

collected tunnel surface images with high accuracy. The experimental results show that the proposed method obtains a better defect detection performance compared with the classic traditional methods and representative instance-aware segmentation methods under complex conditions. In addition, the algorithm to merge the detected defect regions can successfully find each connected defect region for further engineering evaluation, e.g., defect risk grade evaluation, tunnel damage evaluation and tunnel stability evaluation. Our future work will be to extend our method to many other surface inspection applications.

## CRediT authorship contribution statement

**Yingying Xu:** Conceptualization, Methodology, Software, Writing - original draft. **Dawei Li:** Resources, Supervision, Writing - review & editing. **Qian Xie:** Investigation, Formal analysis, Validation. **Qiaoyun Wu:** Validation, Data curation. **Jun Wang:** Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported in part by The National Key Research and Development Program of China (2020YFB2010702), National Natural Science Foundation of China under Grant 61772267, Aeronautical Science Foundation of China (No. 2019ZE052008) and the Natural Science Foundation of Jiangsu Province under Grant BK20190016.

## References

[1] L. Yang, E. Fang, Review and developing trend on technology for detecting metro tunnel structure diseases, Urban Rapid Rail Transit (2017).
[2] L. Zhang, X. Wu, M.J. Skibniewski, J. Zhong, Y. Lu, Bayesian-network-based safety risk analysis in construction projects, Reliab. Eng. Syst. Saf. 131 (2014) 29–39.
[3] J. Wang, X. Xie, H. Huang, A fuzzy comprehensive evaluation system of mountain tunnel lining based on the fast nondestructive inspection, in: 2011 International Conference on Remote Sensing, Environment and Transportation Engineering, IEEE, 2011, pp. 2832–2834.
[4] X. HU, N. BAI, H. LI, Analysis on tunnel accident on line 1 of Saint Petersburg Metro, Tunnel Construction (4) (2008) 7.
[5] H. Shao, H. Huang, D. Zhang, R. Wang, Case study on repair work for excessively deformed shield tunnel under accidental surface surcharge in soft clay, Chin. J. Geotech. Eng. 38 (6) (2016) 1036–1043.
[6] S. German, I. Brilakis, R. DesRoches, Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments, Advanced Engineering Informatics 26 (4) (2012) 846–858.
[7] W. Zhang, Z. Zhang, D. Qi, Y. Liu, Automatic crack detection and classification method for subway tunnel safety monitoring, Sensors 14 (10) (2014) 19307–19328.
[8] R. Wang, T. Qi, B. Lei, et al., Characteristic extraction of cracks of tunnel lining, Chin. J. Rock Mech. Eng. 6 (2015) 1211–1217.
[9] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[10] X. Wang, X. Chen, Y. Wang, Small vehicle classification in the wild using generative adversarial network, Neural Comput. Appl. (3) (2020) 1–11.
[11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[13] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[14] Q. Xie, Y.K. Lai, J. Wu, Z. Wang, Y. Zhang, K. Xu, J. Wang, MLCVNet: Multi-level context votenet for 3D object detection, 2020.
[15] Q. Xie, D. Li, Z. Yu, J. Zhou, J. Wang, Detecting trees in street images via deep learning with attention module, IEEE Trans. Instrum. Meas. PP (2019) 1, http://dx.doi.org/10.1109/TIM.2019.2958580.
[16] Q. Xie, R. Oussama, Y. Guo, M. Wang, M. Wei, J. Wang, Object detection and tracking under occlusion for object-level RGB-D video segmentation, IEEE Trans. Multimed. 20 (2017) 580–592, http://dx.doi.org/10.1109/TMM.2017.2751965.
[17] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[18] G. Ghiasi, C.C. Fowlkes, Laplacian pyramid reconstruction and refinement for semantic segmentation, in: European Conference on Computer Vision, Springer, 2016, pp. 519–534.
[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[20] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[21] X. Du, T.Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q.V. Le, X. Song, Spinenet: Learning scale-permuted backbone for recognition and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11592–11601.
[22] Y. Lee, J. Park, Centermask: Real-time anchor-free instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13906–13915.
[23] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, et al., Resnest: Split-attention networks, 2020, arXiv preprint arXiv:2004.08955.
[24] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[25] C.Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: Deconvolutional single shot detector, 2017, arXiv preprint arXiv:1701.06659.
[26] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: Towards accurate region proposal generation and joint object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 845–853.
[27] T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[28] I. Sobel, G. Feldman, A 3x3 isotropic gradient operator for image processing, 1968, pp. 271–272, a talk at the Stanford Artificial Project in.
[29] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (6) (1986) 679–698.
[30] S. Konishi, A.L. Yuille, J.M. Coughlan, S.C. Zhu, Statistical edge detection: Learning and evaluating edge cues, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 57–74.
[31] G. Bertasius, J. Shi, L. Torresani, Deepedge: A multi-scale bifurcated deep network for top-down contour detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4380–4389.

[32] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.
[33] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3982–3991.
[34] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[35] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[36] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[37] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A review on deep learning techniques applied to semantic segmentation, 2017, arXiv preprint arXiv:1704.06857.
[38] H.w. Huang, Q.t. Li, D.m. Zhang, Deep learning based image recognition for crack and leakage defects of metro shield tunnel, Tunnelling and Underground Space Technology 77 (2018) 166–176.
[39] Q. Luo, X. Fang, L. Liu, C. Yang, Y. Sun, Automated visual defect detection for flat steel surface: A survey, IEEE Trans. Instrum. Meas. (2020).
[40] X. Huang, Z. Liu, X. Zhang, J. Kang, M. Zhang, Y. Guo, Surface damage detection for steel wire ropes using deep learning and computer vision techniques, Measurement (2020) 107843.
[41] Z. Fan, Y. Wu, J. Lu, W. Li, Automatic pavement crack detection based on structured prediction with the convolutional neural network, 2018, arXiv preprint arXiv:1802.02208.
[42] J. Gao, D. Yuan, Z. Tong, J. Yang, D. Yu, Autonomous pavement distress detection using ground penetrating radar and region-based deep learning, Measurement (2020) 108077.
[43] S. Chanda, G. Bu, H. Guan, J. Jo, U. Pal, Y.C. Loo, M. Blumenstein, Automatic bridge crack detection–a texture analysis-based approach, in: IAPR Workshop on Artificial Neural Networks in Pattern Recognition, Springer, 2014, pp. 193–203.
[44] D. Kang, Y.J. Cha, Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging, Comput.-Aided Civ. Infrastruct. Eng. 33 (10) (2018) 885–902.
[45] C. Feng, M.Y. Liu, C.C. Kao, T.Y. Lee, Deep active learning for civil infrastructure defect detection and classification, in: Computing in Civil Engineering 2017, 2017, pp. 298–306.
[46] D. Tabernik, S. Šela, J. Skvarč, D. Skočaj, Segmentation-based deep-learning approach for surface-defect detection, J. Intell. Manuf. 31 (3) (2020) 759–776.
[47] F.C. Chen, M.R. Jahanshahi, NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion, IEEE Trans. Ind. Electron. 65 (5) (2017) 4392–4400.
[48] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, C. Loupos, Deep convolutional neural networks for efficient vision based tunnel inspection, in: 2015 IEEE International Conference on Intelligent Computer Communication and Processing, ICCP, IEEE, 2015, pp. 335–342.
[49] H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning, IEEE Transactions on Medical Imaging 35 (5) (2016) 1285–1298.
[50] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[51] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1991) 583–598.
[52] S. Kamdi, R. Krishna, Image segmentation and region growing algorithm, Int. J. Comput. Technol. Electron. Eng. 2 (1) (2012) 103–107.
[53] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring r-cnn, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6409–6418.