DOI: 10.1111/mice.12577

ORIGINAL ARTICLE

Attention-guided multiscale neural network for defect detection in sewer pipelines

Yanfen Li [1ℓ] | Hanxiang Wang[1ℓ] | L. Minh Dang [2] | Hyoung-Kyn Song[2] | Hyeonjoon Moon [1*]

[1] Department of Computer Science and Engineering, Sejong University, Seoul, Republic of Korea

[2]Department of Information and Communication Engineering, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul, Republic of Korea

[ℓ] These authors contributed equally to this work and should be considered co-first authors.

***Correspondence**
Sejong University, Seoul, Republic of Korea
Email: hmoon@sejong.ac.kr

Funding information

ABSTRACT

Sanitary sewer systems are major infrastructures in every modern city, which are essential in protecting water pollution and preventing urban waterlogging. Since the conditions of sewer systems continuously deteriorate over time due to various defects and extrinsic factors, early intervention on the defects is necessary to prolong the service life of the pipelines. However, prior works for defect inspection are limited by accuracy, efficiency, and economic cost. In addition, the current loss functions in object detection approaches are unable to handle the imbalanced data well. To address the above drawbacks, this paper proposes an automatic defect detection framework that accurately identifies and localizes eight types of defects in closed-circuit television (CCTV) videos based on a deep neural network. First, an effective attention module is introduced and used in the backbone of the detector for better feature extraction. Then, a novel feature fusion mechanism is presented in the neck to alleviate the problem of feature dilution. After that, an efficient loss function that can reasonably adjust the weight of training samples is proposed to tackle the imbalanced data problem (IDP). Also, a publicly available dataset is provided for defect detection tasks. The proposed detection framework is robust against the imbalanced data and achieves a state-of-the-art mAP of 73.4%, which is potentially applied in realistic sewer defect inspections.

1 INTRODUCTION

Underground sewer systems that are utilized to congregate and convey household or industrial wastewater to the treatment facilities have become increasingly critical components in the public infrastructures of modern cities. A well-functional sewer system is capable of ensuring the cleanliness of the human living environment and reducing the spread of epidemics. On the other hand, it plays an important role in drainage and avoiding urban waterlogging in the flood season. Nevertheless, the sewer pipelines with a long service history are inevitably affected by various defects, which seriously restrict their lifespan. Therefore, early detection and intervention of defective regions can effectively prevent the further deterioration of sewer conditions and then save considerable economic costs in the long term. According to the estimation by Bluefield Research in 2020, nationwide expenditures on sewer repair and replacement cost more than $3 billion, addressing over 4,600 miles of pipeline ('2021 Report Card for America's Infrastructure - Wastewater' 2021). With the wide popularization of closed-circuit television (CCTV) in the sewer system, automated defect detection via CCTV cameras has attracted the attention of many researchers from industry and academia in recent years. Compared to the methods with other data acquisition techniques (Khan and Patil 2018; Lepot, Stanić, and Clemens 2017), CCTV inspection approaches are safe, portable, and economical (Mostafa and Hegazy 2021; Czimmermann et al. 2020).

Although the defect detection methods based on computer vision (CV) have gained approved performance (Li et al. 2022; Zhou, Zhang, and Gong 2022), there are still several issues to be resolved. For instance, an excellent CV-based

approach depends on a feature extractor that can effectively extract deep and semantically strong features. But it is challenging for the existing extractors to precisely obtain complete target areas, especially when processing the images with complex backgrounds. To make the model pay more attention to the objective region, a great effort has been put into acquiring more valuable features by deepening and broadening feature extractors in previous studies (Xie et al. 2019; Dang et al. 2021; Wang et al. 2020). However, this strategy which produces more computational burdens, may bring the gradient instability problem, and degrade the model's shallow learning ability (Lu et al. 2017). Recently, different attention modules were designed to focus on object areas and extract sufficient features. Nevertheless, some attentions cannot dynamically calculate the weight for each branch and selectively enhance features in multi-branch models. In addition, some attentions fail to obtain accurate location information from feature maps. As a result, an effective attention is proposed in this study to address the above issues simultaneously.

In addition, the images collected in underground sewer conditions contain many small objects. During the feature extraction process, the feature information of the tiny objects is easy to be mixed with background information when the feature map passes successive convolution and pooling operations (Li, Xie, et al. 2021). The feature pyramid networks (FPN) (Liu et al. 2018; Wu et al. 2020), which directly fuse multi-scales features, are added into object detection models to retain these features. But their fusion methods dilute the semantic information of the features (Luo et al. 2021). The semantic information refers to the discriminative information of the target, such as texture, edge contour and color. After the input image passes through several convolution operations, the position and edge contour information of small objects becomes weaker due to the larger receptive field of the feature map mapped to the original image. Based on prior work, an efficient feature fusion mechanism was proposed to mitigate the problem of feature information decay.

The imbalanced data problem (IDP) mainly covers the uneven distribution of negative (background) and positive (foreground) samples as well as the imbalance between hard and easy samples. Hard / easy samples mean the samples that are hard or easy to be learned and detected by the network. The difficulty of learning a sample is associated with many factors, such as the imbalanced distribution between classes, backgrounds, and sample sizes. The IDP causes the detector to be overwhelmed by dominant samples and influences the model's convergence speed and accuracy (Li, Li, et al. 2021). As for this issue, several loss functions such as focal loss (Lin et al. 2017), generalized focal loss (Li et al. 2020), automated focal loss (Weber, Fürst, and Zöllner 2020), and varifocal loss (Zhang et al. 2021) were carried out to adjust loss values for different samples by hyperparameters. Although the latest varifocal loss can deal with the IDP effectively, it still has the issues of slow convergence speed and overfitting. Because varifocal loss cannot assign proper loss values to different samples. On this basis, the loss equation for the defect detection model is further optimized in this paper.

In view of above limitations, the main contributions of this work are summarized as follows.

- *Design an effective attention module to improve the feature extraction ability.*
- *Introduce an innovative feature fusion mechanism to alleviate the problem of feature dilution in the multi-scales feature fusion.*
- *Propose an efficient loss function to solve the imbalance of training samples.*
- *Provide a manually validated and annotated dataset for defect detection tasks.*

The rest of this article is arranged as follows. Section 2 provides a review of the literature associated with defect detection. The overall flowchart of the proposed system and the corresponding methodology is explained in Section 3. Section 4 describes the data acquisition and annotation processes. After that, several experiments are conducted in Section 5 to demonstrate the contributions of this study. In the end, a conclusion was drawn by indicating current weaknesses and future research directions in Section 6.

2 RELATED WORK

Currently, research involving vision-based defect detection has conveniently been assisting sewer inspectors in evaluating conditions (Zhang and Lin 2022). Many researchers have utilized deep learning (DL) algorithms to facilitate defect inspection due to their automatic feature extraction and subsampling. For example, a faster region-based convolutional neural network (faster R-CNN) was applied to identify the specific type and gain the exact location, and it achieved a mean Average Precision (mAP) of 83% for 4 classes by adjusting different impact factors (Cheng and Wang 2018). Chen et al. put forward a cost-sensitive defect detection network that can minimize the misclassification costs during the learning process (Chen et al. 2019). More recently, Yin et al. employed YOLOv3 for real-time detection and obtained a mAP of 85.37% for six classes of defects. Also, the detector developed in (Yin et al. 2020) was used as an automated labeling tool in a sewer video interpretation system (Yin et al. 2021). A detector-focused architecture is presented to learn sewer pipe defects and properties, which showed excellent performance on multiple tasks (Haurum et al. 2022). Wang et al. devote efforts to evaluate defect severity by detecting and segmenting defects in CCTV images. Faster R-CNN was superior to the other models in the detection agent, but it was confused to the defects with similar shapes or colors (Wang, Luo, and Cheng 2021). Since it is easy for the above methods to ignore information of the small defects, a strengthened region proposal network (SPRN) for defect localization and

fine-grained recognition are introduced to focus on the small objects by fusing local and global features (Li, Xie, et al. 2021). Even though their proposed model was capable of obtaining more contextual information, this feature fusion scheme led to a weak detection performance due to the dilution of semantic information.

Considering that the existing detectors cannot extract adequate features at precise target areas, some researchers deepen and broaden network structures to improve the efficiency of feature extraction (Xie et al. 2019; Dang et al. 2021; Wang et al. 2020). In the latest works, the idea that combines attention modules with detectors shows superior performance due to the strengths of fewer computing resources and better overall learning capacity. For instance, Ban et al. integrated an attention module in a detector to pay attention to the crucial pixels (Ban, Tian, and Zhao 2020). Similarly, Zhu et al. employed a cascade attention module to refine objective regions, and their developed network showed excellent learning ability on salient areas (Zhu et al. 2018). Inspired by those successful approaches, this research attempt to design a customized attention module for the proposed defect detection framework.

Despite rapid advances in DL-based defect detection, the overall detection accuracy has not been considerably improved because of the imbalanced data. As for the IDP, a direct and effective solution is to propose a robust loss function that reasonably adjusts the training weight of different samples. Lin et al. proposed focal loss (FL) to address both foreground-background imbalance and hard-easy imbalance (Lin et al. 2017). According to the evaluation results, a one-stage detector (RetinaNet) with FL outperformed two-stage detectors on detection accuracy and speed. Afterward, several improved versions of FL were proposed. For example, the generalized focal loss (GFL) was presented to optimize the representations of the original FL (Li et al. 2020). Besides, an automated focal loss (AFL) was used to resolve the IDP in object detection tasks by controlling the model's focuses automatically (Weber, Fürst, and Zöllner 2020). Also, the latest Varifocal loss (VFL) added the intersection over union (IoU) into the calculation of loss values to enhance learning signals of positive samples and remain the original calculation equation of FL for negative samples (Zhang et al. 2021). Since the VFL does not involve the probabilities of predicted positive samples, the probabilities and IoU are integrated to redefine a new loss equation in this paper.

3 METHODOLOGY

The overall flowchart of the introduced defect detection framework is presented in Figure 1 (a). Firstly, the CCTV crawler is utilized to collect pipe videos in the groundwater pipeline, and then all frames containing defects are extracted from the acquired videos. The extracted images and the annotation files labelled by LabelMe are divided into training and testing sets. The training images are fed into the proposed Pipe-VarifocalNet (Pipe-VFNet), and the test images are used to evaluate the model's performance. As illustrated in Figure 1 (b), Pipe-VFNet is proposed by improving the VFNet network, which consists of the backbone, neck, and head sections. In this paper, an attention module (CSA) is designed to create a new backbone network (E-ResNeSt) for better feature extraction (Section 3.1). And then, a novel feature fusion mechanism is introduced to be used for the neck (Section 3.2). In addition, an efficient loss function (EFL) is presented in the head (Section 3.3) to handle the IDP.

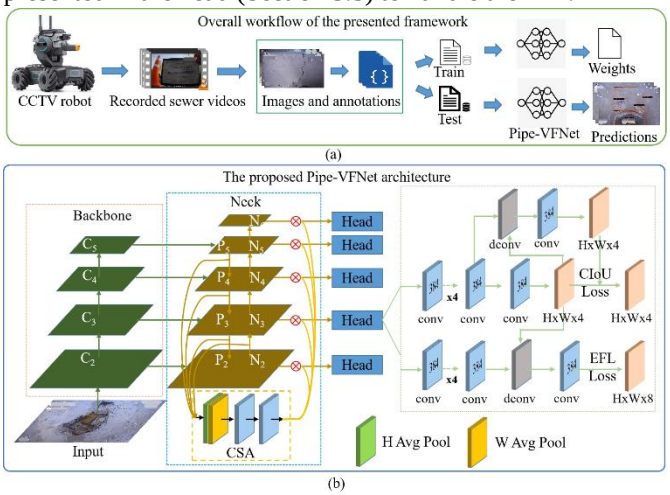


FIGURE 1. (a) Overall flowchart of the presented defect detection framework. (b) The proposed Pipe-VFNet architecture.

3.1 Feature extraction module

The feature extraction module is the most critical part of the deep neural network, which directly affects the model's accuracy and training speed. Recently, the residual networks have shown strong stability in different tasks such as classification (Lu et al. 2020), detection (Cai and Vasconcelos 2019), and segmentation (Wang et al. 2021). In this study, the performances of four excellent residual modules (ResNet, ResNeXt, Res2Net, and ResNeSt) are compared, and the most suitable module with the best performance is then selected as the feature extractor (backbone section) to construct the defect detection network. Also, the depths of these models are uniformly set to 101 for an impartial comparison based on the same dataset. The following section gives a concise and clear introduction to these four residual modules and a proposed residual module.

3.1.1 ResNet

ResNet is the first residual network proposed in (He et al. 2016) and outperformed other networks in the ILSVRC2015 competition. The architecture of ResNet is composed of several consecutive residual blocks. As shown in Figure 2 (a), each residual block contains two 1x1 convolution layers, a 3x3 convolution layer, and a shortcut structure. Residual blocks can protect the integrity of information during the transmission process to avoid gradient disappearance or explosion, and it is capable of reducing the over-fitting probability and accelerating the model's convergence in the training process. Moreover, the design of shortcut structure makes the network only focus on the differences between input and output in the whole learning process, which reduces the learning difficulty of the model. Based on the above advantages, ResNet is increasingly used in recent object

detection research (Cai and Vasconcelos 2019; Cao, Cholakkal, et al. 2020; Lin et al. 2017).

3.1.2 ResNeXt and Res2Net

ResNeXt (Xie et al. 2017) and Res2Net (Gao et al. 2019) are proposed to optimize ResNet by adopting the idea of stacking layers and the structure of grouped convolutions. As shown in Figure 2 (b), ResNeXt uses the split-transform-merge strategy to extract and fuse features. Feature maps are divided into 32 groups in each residual structure, and then, the features generated from each group are merged and sent to a 1x1 convolution layer. This method can improve the model's learning ability while ensuring that the complexity of the model is basically unchanged.

Unlike ResNeXt, Res2Net improves the model's receptive field by adding small residual units in the basic residual block. Figure 2 (c) shows that the features extracted from the first 1x1 convolution layer are divided into four parts (X1, X2, X3, and X4) in the residual block of Res2Net. Except for the X1 part, the feature maps in other parts pass a 3x3 convolution operation. Prior to the convolution operation, X3 and X4 combine with Y2 and Y3, respectively. After that, four outputs (Y1, Y2, Y3, and Y4) containing different receptive fields are fused and sent to the second 1x1 convolution layer. In this way, the model learns the feature information comprehensively by fusing the feature maps with multiple receptive fields.

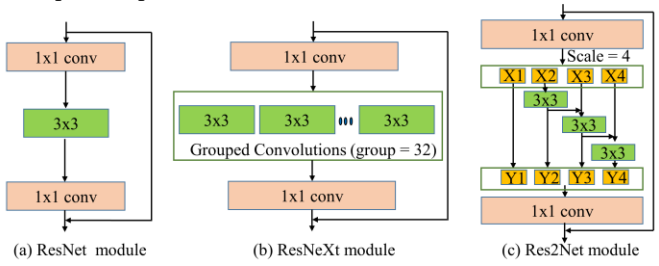


FIGURE 2. The structures of three different feature extraction modules. (a) ResNet module, (b) ResNeXt module, (c) Res2Net module.

3.1.3 ResNeSt and Enhanced ResNeSt (E-ResNeSt)

Based on the original structure of ResNet, the ResNeSt (Zhang, Wu, et al. 2020) network uses a multi-branch structure to enhance the model's diversified expression ability and adopt the attention module to concentrate on the objective region. As shown in Figure 3 (a), two hyperparameters (cardinality (K) and radius (R)) are used in ResNeSt to embody the multi-branch idea. ResNeSt first divides all the input features into K groups, and then each group of features is divided into R sub-groups. In this study, K and R are set to 16 and 4, respectively. After that, feature maps of each sub-group are successively input to a 1x1 convolution layer and a 3x3 convolution layer. The output features of all sub-groups (R sub-groups) in each group are fused together and sent to the split attention module. In the end, the output features of all groups (K groups) are also fused and input into the 1x1 convolution layer. The split attention module can redistribute the weight between feature channels by obtaining relevant features in channels. As illustrated in Figure 3 (b), the output features of all sub-groups are fused and input to the global average pooling layer for squeezing information in the channel direction. Equation (1) represents the calculation process in the global average pooling layer, it calculates the output features F_n of the n^{th} channel after squeezing the input features X. n is the channel index, and the range of the n depends on the channel number of the input feature tensor. H and W are the height and width of the input feature. x and y represent coordinate positions. After a convolution layer, a batch normalization (BN) layer, and a ReLU activation function, feature maps are divided into R groups and input into a 1x1 convolution layer. The softmax function is used to compute the attention weight coefficient of each sub-group. The input features from different sub-groups are multiplied by the corresponding attention weight coefficients and then merged to generate new feature maps.

$$F_n = \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} X_n(x, y) \quad (1)$$

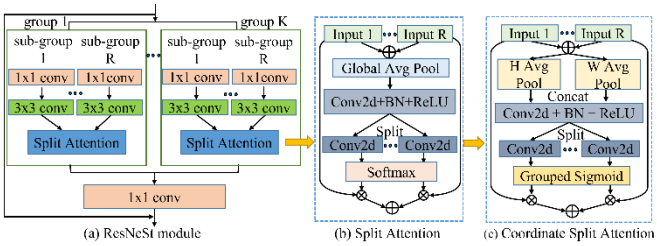


FIGURE 3. (a) The architecture of ResNeSt module, (b) the structure of the split attention, and (c) the structure of the proposed coordinate split attention.

The design of the split attention module is similar to that of the SE attention module (Hu, Shen, and Sun 2018), which uses two-dimensional global pooling to calculate channel attention. Although this method reduces the computational burden of the network, it only concentrates on the channel information and loses the location details in the calculation process (Hou, Zhou, and Feng 2021). To address the problem, an improved attention module called coordinate split attention (CSA) is introduced in the proposed Enhanced ResNeSt module (E-ResNeSt) to replace the split attention module in ResNeSt, as shown in Figure 3 (c). The novelty of the presented CSA module is to dynamically acquire the adequate objective location information from multiple inputs by integrating a grouped sigmoid function with the one-dimensional global pooling operation proposed in (Hou, Zhou, and Feng 2021). Instead of using single two-dimensional global pooling to extract all features, two parallel one-dimensional global pooling operations respectively generate the features containing spatial data in vertical and horizontal directions. The two global pooling kernels are (H,1) and (1, W). Equation (2) and (3) represent the output features F_n^h and F_n^w of the n^{th} channel after the squeeze of the input features X with height H and width W, respectively. This squeeze approach enables the attention module to capture the location information of the region of interest accurately. After that, the feature tensors from two directions are concated and fed into the block that includes the convolution, BN, and ReLU operations. Then, the

concatenated feature tensor containing the spatial information is sent to the following convolutional layers for feature selection. The selected features are represented in attention map groups corresponding to input tensors. Finally, the grouped sigmoid function is used to calculate the attention weight coefficients in the horizontal and vertical directions by the generated attention maps for each input. The process of attention module is shown in Equation (4). After fusing all groups of feature vectors, a channel shuffle algorithm is added into the residual block to facilitate the information flow between groups. Channel shuffle further improves the model's feature representation and generalization abilities.

$$F_n^h = \frac{1}{W}\sum_{i=1}^{W} X_n(h,i) \quad (2)$$

$$F_n^w = \frac{1}{H}\sum_{i=1}^{H} X_n(j,w) \quad (3)$$

3.2 Feature fusion mechanism

An efficient feature fusion mechanism is of great significance for the detectors to strengthen learning ability. In the feature maps extracted from the backbone network, the low-level features contain accurate target location and less semantic feature information. On the contrary, the high-level features contain rich semantic features and rough target location information. Path Aggregation Network (PAFPN) (Liu et al. 2018) is capable of enhancing the feature expression ability of the backbone section by extracting and fusing these feature maps with different scales. As shown in Figure 4 (a), the PAFPN structure includes a top-down path and a bottom-up path for feature reinforcement. The top-down path shown by blue arrows performs up sampling on the feature maps to make full use of the semantic information in the high-level features. The bottom-up path represented by orange arrows aims to acquire the network's low-level feature information. Because the shallow features contain lots of object contour information, this information is more conducive to the model to locate the object position. The PAFPN realizes the information complementarity between high-level features and low-level features, but some semantic features are diluted in the process of feature fusion at different scale levels. Moreover, the direct fusion of semantic features with multi-scales may lead to aliasing effects (Luo et al. 2021).

To alleviate the above problems, a feature pyramid structure (Att-PAFPN) that adopts a new feature fusion mechanism is proposed by combining it with an attention module. As shown in Figure 4 (b), four different scale features {C_2, C_3, C_4, C_5} extracted by backbone are input into Att-PAFPN. The features {P_2, P_3, P_4, P_5} are generated after 1x1 convolution operation and the top-down path of Att-PAFPN. All features are fused together and input into the proposed CSA module, which can fully utilize rich semantic information and alleviate aliasing effects. Then the CSA calculates the attention weight coefficient from the fused feature map (M). After that, each output feature is corrected by multiplying the attention weight coefficient. Equation (4) shows the calculation of the attention weight coefficient in the CSA module, and Equation (5) represents the correction process of output features.

$$CSA(x) = \sigma\big(f([HAvgPool(x), WAvgPool(x)])\big) \quad (4)$$

$$y_i = CSA(M) \times N_i \quad (5)$$

where x is the input feature, $HAvgPool$ and $WAvgPool$ are the global pooling operation in vertical and horizontal directions respectively. f indicates the convolution operation, and σ means the grouped sigmoid function. N_i are the output features {N_2, N_3, N_4, N_5, N_6} generated from the bottom-up path and P_i. For example, N_2 is directly obtained from P_2, N_3 is from P_3 and N_2. The y_i are the final output features after the correction process.

The BN layer in PAFPN normalizes the data with batch size as the dimension, and the value of batch size affects the model's performance during training. In this study, a small batch size (2) is set to adapt the limited computing memory. Nevertheless, the model's gradient values may become unreliable and fluctuate seriously when the batch size is set to a small value. Therefore, group normalization (GN) is used to prevent the Att-PAFPN section from the effect caused by batch size (Wu and He 2018). In addition, another modification is implemented by increasing the output channels of the PAFPN for further improvement.

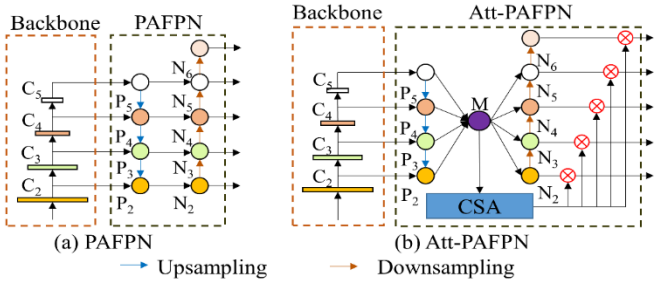


FIGURE 4. (a) The architecture of PAFPN, and (b) the architecture of the proposed Att-PAFPN.

3.3 Loss function

There are two loss functions in the existing object detection models, including the classification loss and the location regression loss. The classification loss is utilized to calculate the error between the predicted class and the Ground Truth (GT) class so as to update the network parameters. In sewer defect datasets, a great quantity of easy negative samples inevitably overwhelm detection networks. That makes the model pay much attention to the easy negative samples and ignore the learning of hard samples. Focal loss (FL) (Lin et al. 2017) adds two hyperparameters (α and γ) into the cross-entropy loss to adjust the weights of positive-negative samples and hard-easy samples, which reduces the influence of easy negative samples on the detector. As shown in Equation (6), p represents the probability value that the object in the anchor box is predicted to be a positive sample, and q is used to judge whether the prediction is consistent with the GT. However, the FL function should not use the hyperparameter γ to down-weight both negative and positive samples because positive samples are more important and rarer than negative samples.

$$FL(p,q) = \begin{cases} -\alpha(1-p)^{\gamma}\log(p) & if\ q=1 \\ -(1-\alpha)p^{\gamma}\log(1-p) & else \end{cases} \quad (6)$$

Equation (7) indicates the calculation of the varifocal loss (VFL) (Zhang et al. 2021). To retain the learning signal of positive samples, the VFL removes the hyperparameter γ when it calculates the loss from positive samples and introduces intersection over union (IOU) values ($q^* \in [0,1]$) between the prediction and the GT. Compared with the FL, the VFL obtain higher accuracy but slow convergence speed. This is because the VFL pays excessive attention to easy samples when processing positive samples.

$$VFL(p,q^*) = \begin{cases} -q^*(q^*\log(p) + (1-q^*)\log(1-p)) & if\ q^*>0 \\ -\alpha p^{\gamma}\log(1-p) & else \end{cases} \quad (7)$$

Inspired by the VFL, an efficient FL (EFL) shown in Equation (8) is proposed to deal with the above issues. The probability (p) predicted as positive samples and the IOU values ($q^* \in [0,1]$) are considered in the calculation of the EFL. When a positive sample obtains the high probability, the loss calculated by FL is reduced to a value close to 0, which means the model almost ignores the learning for easy positive samples. In some cases, VFL gives a large loss value to the easy positive sample and assigns a small loss value to the hard positive sample. The instable calculation of VFL leads to inferior convergence speed and accuracy of the model. In contrast, EFL allows the model to keep learning these easy positive samples without excessive effort because it can appropriately decrease the loss values via a smooth calculation method. Let's suppose that the values of p and q^* are 0.8, the loss values calculated by FL, VFL and EFL are 0.002, 0.40 and 0.01, respectively. It is obvious that EFL can provide a more proper loss value than the others.

$$EFL(p,q^*) = \begin{cases} -(1-p)(1-q^*)\log(p) & if\ q^*>0 \\ -\alpha p^{\gamma}\log(1-p) & else \end{cases} \quad (8)$$

Different from the classification loss, the location regression loss is calculated by comparing the coordinate information of the predicted bounding box (BB) and the actual BB. The location regression loss used in this work is defined in Equation (9).

$$L_{regression} = 1 - IoU + \frac{d^2(A,B)}{l_C^2} + \rho\sigma \quad (9)$$

where A and B indicate the predicted BB and the actual BB, respectively. C is the minimum enclosing rectangle of A and B.

$$IoU = \frac{area(A)\cap area(B)}{area(A)\cup area(B)} \quad (10)$$

IoU is shown in Equation (10). $d(A,B)$ is the Euclidean distance between the center points of A and B. l_c is the diagonal length of C. σ is used to reflect the similarity of the aspect ratio of A and B, and its calculation method is expressed in Equation (11).

$$\sigma = \frac{4}{\pi^2}(arctan\frac{w^B}{h^B} - arctan\frac{w^A}{h^A})^2 \quad (11)$$

w and h are width and height of the BB. ρ is the weight coefficient used to balance the loss function, as shown in Equation (12).

$$\rho = \frac{\sigma}{(1-IoU)+\sigma} \quad (12)$$

Algorithm 1 is proposed to calculate the loss in the entire defect detection network. Each iteration of the model generates the prediction and target. The prediction stores the predicted BB and the classification score (p) of the anchor box. And the target stores the IoU score (q^*) and actual BB. The BB is in a form of (x_1, y_1, x_2, y_2). Firstly, p, q^*, A, B are obtained from the prediction and target tensor. A and B indicate the predicted BB and actual BB, respectively Then, the classification loss and location regression loss are calculated with the pre-defined hyperparameters for the prediction result. Finally, the two values (L_{class} and $L_{regression}$) are combined by a weight coefficient λ.

Algorithm 1: Loss Function of the proposed Network

Input: *prediction, target.*

Output: $Optimized\ Loss$

1. **Initialize:** $\alpha = 0.75,\ \gamma = 1.5,\ \lambda = 2,\ L_{class} = 0,\ L_{regression} = 0,\ Loss = 0$
2. Obtain p, q^*, A, B from prediction and target
3. **If** $q^* > 0$ **then**
4. $\quad L_{class} = -(1-\mathrm{p})(1-q^*)\log(p)$
5. **Else**
6. $\quad L_{class} = -\alpha p^{\gamma}\log(1-p)$
7. **End If**
8. Finding the smallest box C containing A and B.
9. $IoU = \frac{\text{Intersection of area A and B}}{Union\ of\ \text{area}\ A\ and\ B}$
10. Calculating the overlap between box A and box B:

$$\text{Overlap} = \frac{d^2(A,B)}{l_C^2}$$

11. Calculating the similarity of the aspect ratio between A and B using Equation (12).
12. Calculating the regression loss $L_{regression}$ using Equation (9).
13. $Loss = L_{class} + \lambda L_{regression}$

4 SEWER DEFECT DATASET

The CCTV inspection videos were acquired in Seoul, Korea, by the Civil Engineering and Building Technology institute. In the original sewer defect dataset, a total of 4,383 images with 5,385 distinct defects are extracted and validated from the sewer videos in a manual manner. Therefore, each frame is independent and distinct in the collected data. Eight classes

of common defects are considered as the detection targets in this research, which include debris silty, horizontal crack, vertical crack, joint faulty, joint open, lateral protruding, pipe broken, and surface damage. Figure 5 shows eight types of example images from the proposed dataset. The entire data is first randomly separated into training and testing sets according to the ratio of 8:2. And then the original training set is expanded by different data augmentation methods, such as cutout, gaussian blur, and channel shuffle. The detailed information of eight classes before and after data augmentation is displayed in Table 1.

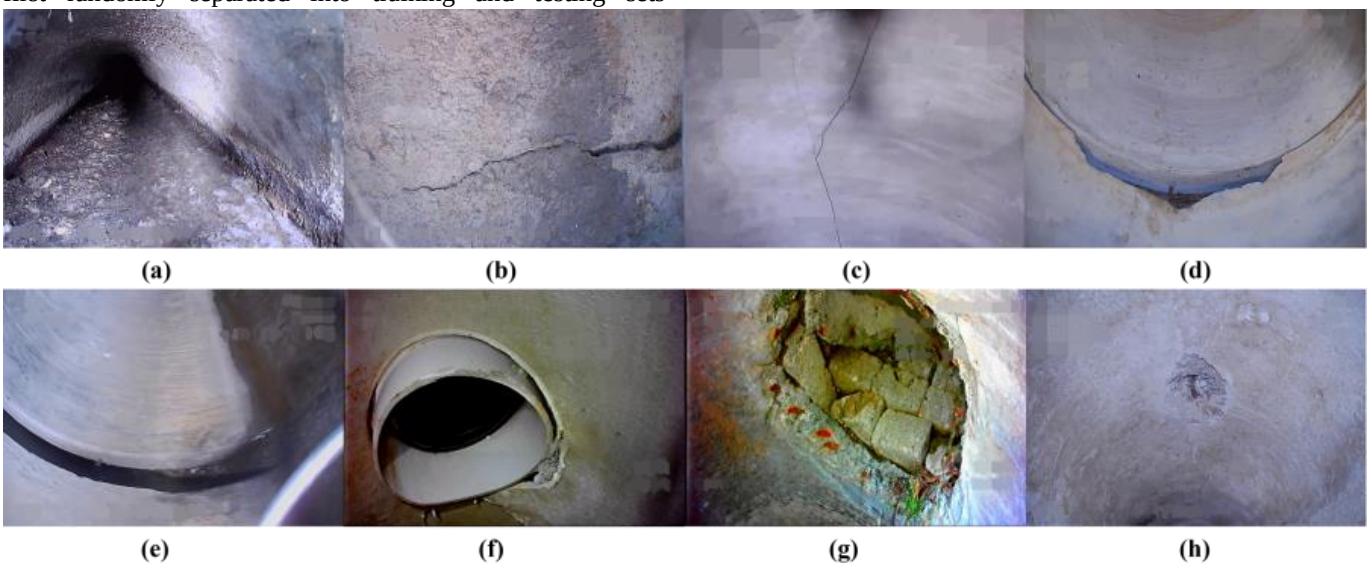


FIGURE 5. Examples of eight types of defects. (a) debris silty, (b) horizontal crack, (c) vertical crack, (d) joint faulty, (e) joint open, (f) lateral protruding, (g) pipe broken, and (h) surface damage.

TABLE 1. The sample numbers in the proposed sewer defect dataset before and after data augmentation.

Defect	Before data augmentation	After data augmentation
Debris silty	395	1,031
Horizontal crack	711	1,843
Vertical crack	709	1,852
Joint faulty	770	1,999
Joint open	652	1,698
Lateral protruding	1,020	2,652
Pipe broken	657	1,711
Surface damage	471	1,224
Total	5,385	14,010

5 EXPERIMENTAL RESULTS

5.1 Feature extraction

In this section, a comparison with some state-of-the-art (STOA) backbones was conducted to demonstrate the significance of the proposed feature extraction model (backbone section) on the collected dataset. Table 2 lists the performances of different backbone models on the base of the original VFNet framework and the PAFPN neck section from the aspects of network parameters (Param.), computations (FLOPs), and detection precision (mAP). The floating point operations per second (FLOPs) is a widely used indicator to measure the computational complexity of the model. The entire number of network operations that can be summed up into a single floating-point hardware operation is the definition of FLOPs (Molchanov et al. 2016; Langerman et al. 2020). The original ResNeSt is much lighter than Res2Net due to fewer parameters (37.28M) and computations (211.15G). Compared with ResNeSt, E-ResNeSt has 0.14M more parameters and 0.24G more FLOPs because it was improved based on ResNeSt by adding a novel attention module. On the other hand, the proposed E-ResNeSt module achieved the highest mAP of 69.1%, which is 1.4% higher than the original ResNeSt. It reflects that the designed backbone model has superior feature extraction capability for sewer defect images.

TABLE 2. Defect detection performances of different backbone models. Note: Param. indicates the network parameters, floating-point operations (FLOPs) represent the computations, and mAP is the mean Average Precision of detection.

Backbones	Neck	Framework	Param.	FLOPs	mAP	mAP_{50}	mAP_{75}
ResNet	PAFPN	VFNet	56.93M	281.77G	64.2	86.4	72.5
ResNeXt			56.56M	285.69G	65.5	88.7	74.8
Res2Net			57.59M	292.78G	66.1	90.3	75.6
ResNeSt			**37.28M**	**211.15G**	66.7	91.5	77.2
E-ResNeSt (Proposed)			37.42M	211.39G	**69.1**	**92.3**	**78.9**

Moreover, the feature extraction processes of ResNeSt and E-ResNeSt are visualized and compared to highlight the improvement of E-ResNeSt. There are four stages in these two backbone models, which are used to generate feature maps {C_2, C_3, C_4, C_5} and input them into the neck section. The features (C_3) extracted from the second stage contain more information regarding objective regions than other stages, and it could better reflect the model's extraction ability. As a result, the feature maps from the second stage of both backbone models are illustrated in Figure 6. According to the comparison between two experimental models, it is clear that the extracted features of E-ResNeSt are visually diversified and valid owing to a satisfying representation ability. For example, the feature map (line 3 column 3) obtained from the E-ResNeSt provides discriminant details, such as the contour and shape of the protruding pipe.

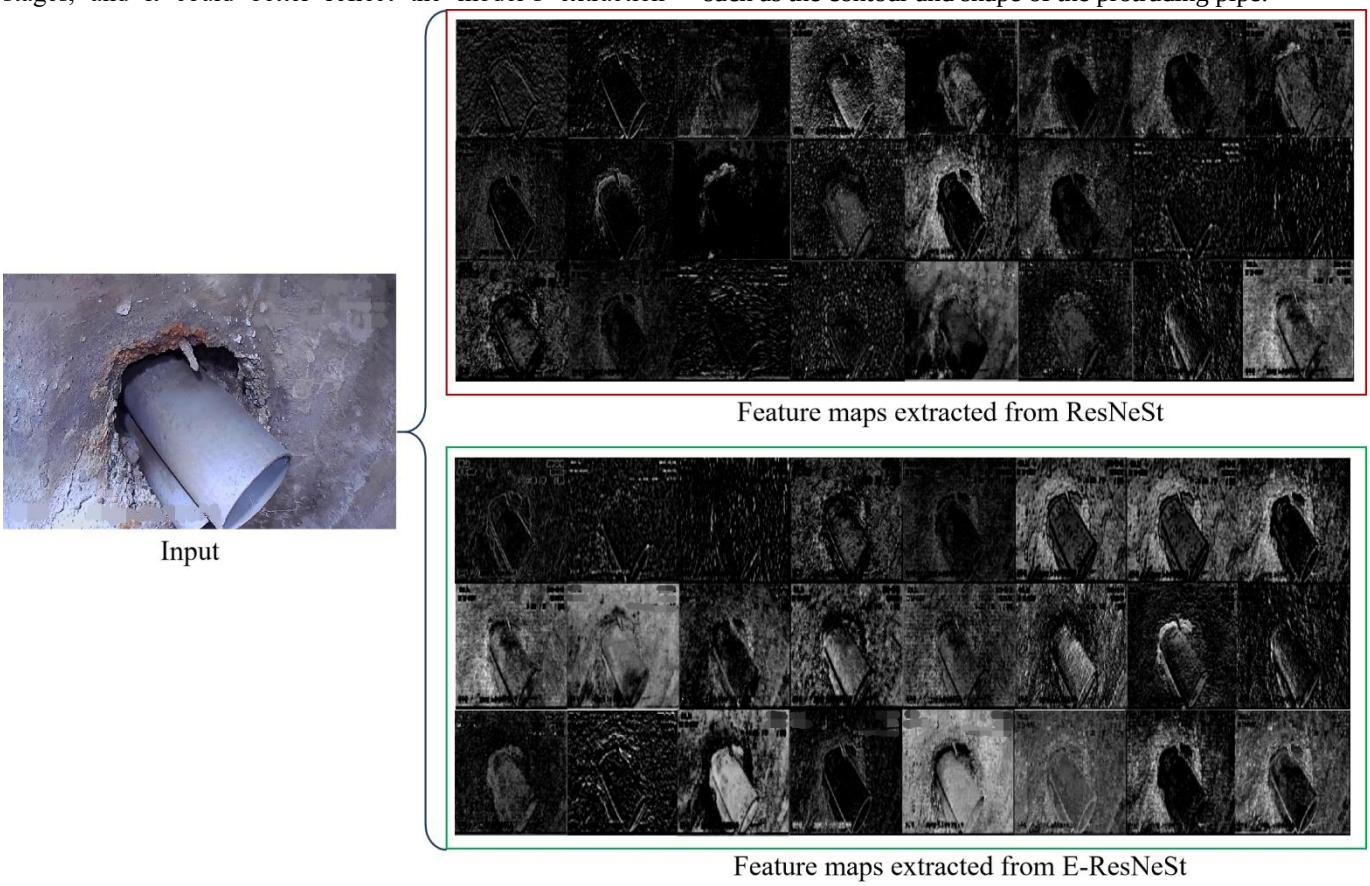


FIGURE 6. The visualized feature extraction processes for ResNeSt and E-ResNeSt.

5.2 Feature fusion

The objective of this experiment is to prove the effectiveness of several modifications to the original PAFPN. Table 3 shows the detection performances of the neck models with distinct configuration settings. The third row contains the main structural components of the raw PAFPN model, and the last row has the main structural components of the Att-PAFPN model. It can be effortlessly realized that the model's mAP considerably increased from 0.691 to 0.704 after adding the proposed attention module. This means the presented attention module is helpful for learning the target areas. Apart from that, the original BN is replaced by GN in the proposed neck models to prevent excessive gradient fluctuation. Besides, the output channels are adjusted from 256 to 384 in order to represent more feature information. After a series of optimization, the neck model achieved the best performances (Precision: 0.92, Recall: 0.935, F1: 0.927, mAP: 0.712).

TABLE 3. Defect detection performances of different neck models with different structure configurations. Note: The performances are evaluated by different metrics (precision, recall, F1, and mAP).

Backbone	Neck			Output channels		Precision	Recall	F1	mAP
	PAFPN	Attention	GN	256	384				
EResNeSt	√			√		0.903	0.907	0.905	0.691
	√	√		√		0.912	0.921	0.916	0.704
	√	√	√	√		0.915	0.926	0.920	0.708
	√	√	√		√	**0.92**	**0.935**	**0.927**	**0.712**

In addition, the effect of the introduced feature fusion approach in the neck section is explored. As mentioned above, the Att-PAFPN is presented by modifying the structure of the original PAFPN in this research. The visualization results from these two approaches are contrasted and shown in Figure 7 to emphasize the improvement of the proposed feature fusion mechanism. The first row is the input image, the middle rows represent the feature maps from five different output layers (PAFPN: $\{N_2, N_3, N_4, N_5, N_6\}$, Att-PAFPN: $\{y_2, y_3, y_4, y_5, y_6\}$), and the last row is the final output image. By comparing the feature maps of two networks, Att-FAFPN obtains more features and learns the defective regions more specific than PAFPN. For the predicted result of PAFPN at the first column, the defect in the yellow circle is not detected, and the defect in the green circle is not fully located. Also, the PAFPN detected two bounding boxes on the same defect (row 7 column 3). Compared with the original PAFPN, the proposed Att-PAFPN precisely localizes and classifies all defects.

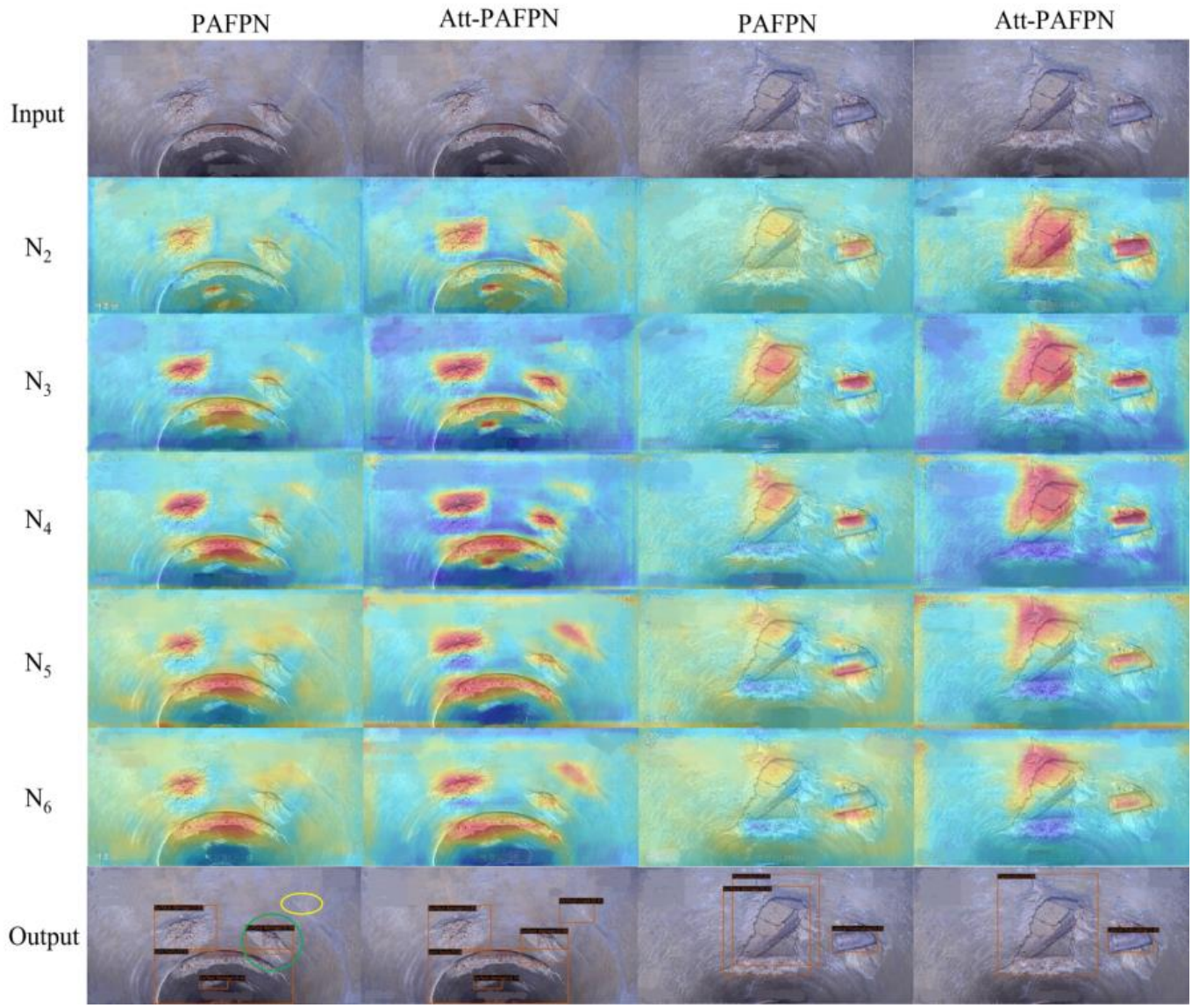


FIGURE 7. Visualization results after the feature fusion process using the original PAFPN and proposed Att-PAFPN.

Since the case of connecting attention and pyramid has been reported for object detection in a recent literature (Cao, Chen, et al. 2020), the performance of the proposed feature fusion mechanism is compared with their presented network. As shown in Table 4, the proposed Att-PAFPN structure is superior to the AC-FPN in terms of the mAP metrics. That implies our Att-PAFPN has a strong feature improvement ability.

TABLE 4. Performances of different neck models connecting attention and pyramid.

Method	Backbone	Neck	mAP	mAP_{50}	mAP_{75}
VFNet	E-ResNeSt	AC-FPN (Cao, Chen, et al. 2020)	0.688	0.919	0.790
		Att-PAFPN	**0.712**	**0.936**	**0.797**

5.3 Loss function

Based on the calculations of the FL and VFL functions, an efficient loss function named EFL was introduced to lighten the IDP and improve the overall detection precision. This experiment aims to testify the impact of the proposed EFL function on the final detection performance. Figure 8 illustrates the Average Precision (AP) for eight classes of defects using three different loss functions. Compared with the vertical crack class, the horizontal crack class that obtained the lower AP value is consider as hard samples. For the horizontal crack class, the AP of the detector with EFL is 16% higher than the AP of the detector with FL. In contrast, the AP of the model with EFL for the vertical crack class decreased 2% compared to that of the model with FL. Similar with the cases of horizontal and vertical cracks, the AP of the model with EFL for the joint open class (hard samples) was improved than the others, while the AP of the model with EFL for the joint faulty class (easy samples) was slightly lower than that of the model with VFL. That means the EFL makes the detector focus more on the harder samples with a lower AP to balance the AP values between hard and easy samples. Taken overall, the model with EFL obtains better AP for most of the classes than other experimental loss functions. In terms of the mAP, the proposed Pipe-VFNet with EFL achieves the highest detection accuracy of 73.4%, which is 4.5% and 2.2% better than the results of FL and VFL.

The data used in above experiment is separated into training and testing sets according to the ratio of 8:2. To investigate how other split ratios between the training set and testing set affect the model's overall performance, the mAP values of the proposed model with different loss functions are computed in Figure 9. It can be observed that the optimal division ratio of training and testing sets is 8:2. Although the training and testing are divided with different ratios, the model with the EFL always outperforms the model with the other experimental loss functions (FL and VFL).

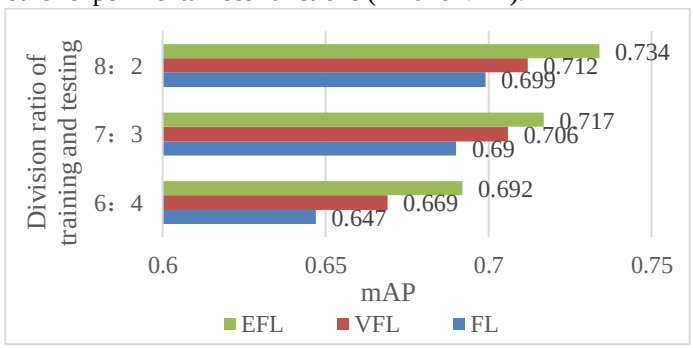
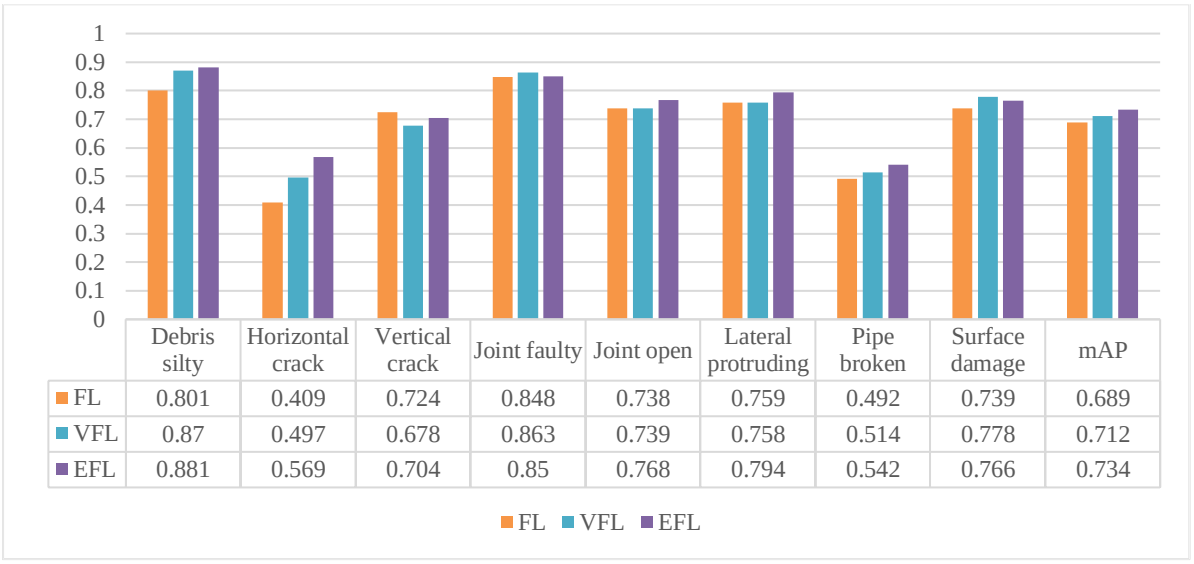


FIGURE 9. mAP values of the proposed model with different loss functions according to different division ratios of training and testing sets.



	Debris silty	Horizontal crack	Vertical crack	Joint faulty	Joint open	Lateral protruding	Pipe broken	Surface damage	mAP
FL	0.801	0.409	0.724	0.848	0.738	0.759	0.492	0.739	0.689
VFL	0.87	0.497	0.678	0.863	0.739	0.758	0.514	0.778	0.712
EFL	0.881	0.569	0.704	0.85	0.768	0.794	0.542	0.766	0.734

FIGURE 8. Average Precision for each class under different loss functions (FL, VFL, and EFL).

Correspondingly, the mAP and loss values are computed and recorded to reflect the training situation in different epochs / iterations. As shown in Figure 10, the blue, orange, and green curves plot the performances of the proposed Pipe-VFNet with FL, VFL, and EFL, respectively. The Pipe-VFNet with EFL has the highest and smoothest curve, which reveals the best detection accuracy (mAP: 73.4%) and the most stable training. By observing the loss curves, the loss values are decreasing continuously with the increase of the iterations. Although the loss of the EFL function decreases slowly in the first 20,000 iterations compared to the loss of the FL, it achieves the smallest loss of 0.13 among the three loss functions. In addition, the models with different loss functions train the same number of iterations at different training speeds. The model with FL obtained the fastest training speed, and its total time for training is 15.4 hours. Since VFL and EFL involve more calculations, the models with these two loss functions require longer training time. The training time of the model with EFL is 16.6 hours, which is similar to that of the model with VFL (16.7 hours).

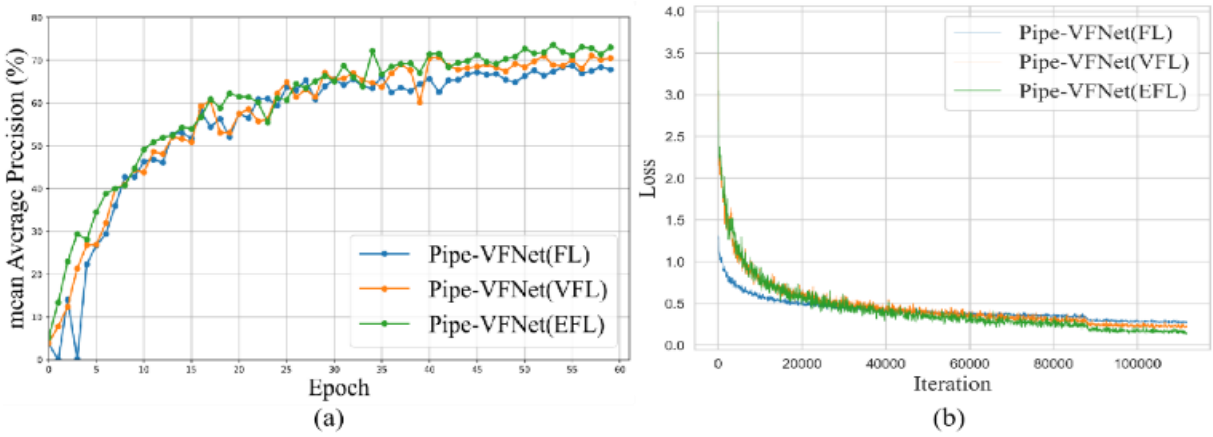


FIGURE 10. The mAP and loss curves of the proposed Pipe-VFNet model using different loss functions.

In order to select the best loss function hyperparameters, the mAP values of the model with different settings are obtained and compared. Three hyperparameters (α, γ, and λ) are included in the proposed loss function. α and γ are determined according to the conducted nine groups of experimental results. For the weight coefficient λ, it was set the same value as the VFNet to balance the classification loss and the regression loss. As shown in Figure 11, the model achieved the highest mAP when α and γ are set to 0.75 and 1.5, respectively.

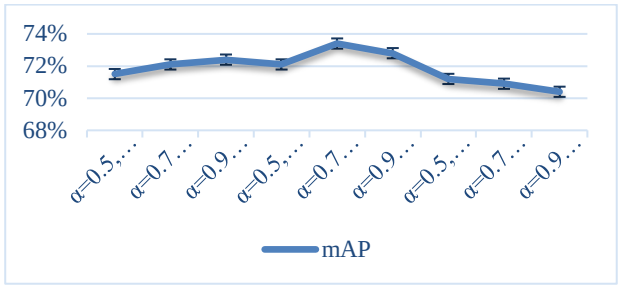


FIGURE 11. The mAP of the proposed Pipe-VFNet model with different hyperparameter settings.

Moreover, the effect of the proposed loss function on our data with different imbalance ratios is tested and reported in Table 5. Two types of sewer images are randomly selected from the acquired data, and different ratios are set for these two classes by adjusting the image number of each class. It can be observed that the Pipe-VFNet model with the proposed loss function is not evidently affected by the IDP when the balancing ratio varies considerably in terms of mAP and loss. For example, the mAP and loss values of Pipe-VFNet declines 3.3% and 0.14 when the balancing ratio changes from 1:1 to 1:10.

TABLE 5. mAP and loss values of the proposed Pipe-VFNet model using the proposed loss function on different balancing ratios. LP refers to lateral protruding, and SD means surface damage.

LP:SD	1:1	1:3	1:5	1:7	1:10
mAP (%)	89.37	89.41	88.79	87.62	86.07
Loss	0.17	0.17	0.19	0.24	0.31

5.4 Model performance

To measure the model's performance comprehensively, the defect classification results in the form of a confusion matrix, and the visualized defect detection results under various conditions are calculated and discussed in this section. Figure 12 shows some instances of defect detection under diverse sewer pipeline conditions. The usual case is that there is only one defect in the same frame extracted from the recorded CCTV videos. Nevertheless, the proposed detector also performs well for the image with multiple defects. As shown in Figure 12 (e), the pipe broken class and the surface damage class are precisely localized and identified with high confidence scores. Even though the confidence scores for both defects in Figure 12 (c) are not high, the model can correctly detect and recognize these defects. Furthermore, the model has satisfactory performances for the defect images with complex backgrounds or bad conditions. For example, Figure 12 (d) adds some additional manual noises, and Figure 12 (f) exists the overexposure problem. The generated promising results demonstrate the constructed detection model is robust against challenging cases.

A confusion matrix is presented in Table 6, and the precision, recall, and F1 score are computed for each type of defect. The proposed defect detection model is performed on the testing set containing 1077 images for eight categories. The bold numbers on the diagonal are close to the total numbers, which indicates the excellent performance of the framework. The model obtains the highest precision of 1.000 on the debris silty class, while the highest recall of 0.977 is obtained for the joint open class. Besides, the F1 score is measured by computing the harmonic average of precision and recall. The proposed model has the best F1 score of 0.987 towards the debris silty class, while the other seven types of defects have an F1 score between 0.859 to 0.966.

TABLE 6. Confusion matrix and F1 score of the testing set. Note: Debris silty (DS), Horizontal crack (HC), Vertical crack (VC), Joint faulty (JF), Joint open (JO), Lateral protruding (LP), Pipe broken (PB), Surface damage (SD).

	DS	HC	VC	JF	JO	LP	PB	SD	Miss	Total	Precision	Recall	F1
DS	**77**	0	0	0	0	0	0	0	2	79	1.000	0.975	0.987
HC	0	**116**	7	2	3	0	0	3	6	137	0.967	0.847	0.903
VC	0	3	**136**	0	1	0	0	1	5	146	0.944	0.932	0.938
JF	0	0	0	**150**	2	0	1	1	1	155	0.955	0.968	0.962
JO	0	1	0	1	**126**	0	0	0	1	129	0.955	0.977	0.966
LP	0	0	0	0	0	**194**	3	4	3	204	0.980	0.951	0.965
PB	0	0	0	2	0	3	**115**	9	3	132	0.950	0.871	0.909
SD	0	0	1	2	0	1	2	**85**	4	95	0.825	0.895	0.859

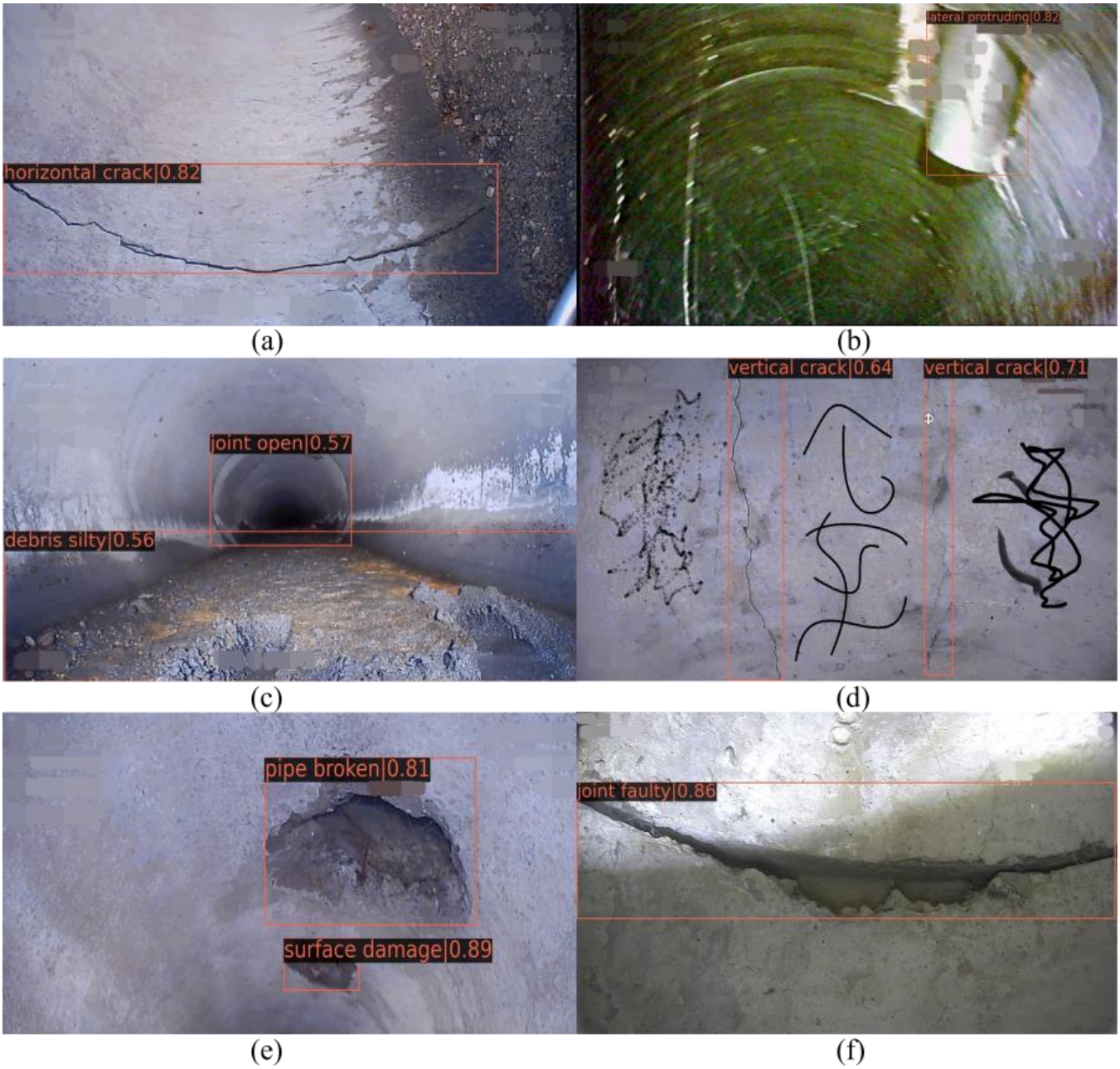


FIGURE 12. Defect detection examples with single defect or multiple defects under different backgrounds.

5.5 Comparison with previous work

In this section, a performance comparison of several recent studies about defect detection is investigated in Table 7. Herein, the proposed Pipe-VFNet model involves a new backbone model (E-ResNeSt), a novel neck model, and a customized loss function. The defect dataset introduced in this study contains the most defect types among the experimental datasets, enabling more exhaustive research. Moreover, the proposed method achieved a detection accuracy of 95.7% in terms of mAP_{50}. Another noteworthy point is that the detection speed of the presented Pipe-VFNet is slower than the YOLOv3 model. That is caused by different computer performances and input sizes. For example, the input size of the YOLOv3 model (Yin et al. 2020) is 416x416, whereas the input size of the constructed model is 1,333x800.

It is challenging to compare different approaches based on different datasets and evaluation metrics. As a result, several advanced algorithms in recent object detection studies are tested in order to provide a fair comparison by implementing on the same dataset (proposed defect dataset) and metrics (AP and mAP). The Faster R-CNN with the SPRN method is not included in this comparison because their implementation details are not available. As shown in Table 8, the AP value for each class and the mAP that represents the mean value of the AP within the scope of 0.5-0.95 IoU thresholds are calculated to evaluate the performances of six different detectors. It can be observed that the proposed model did not achieve the fastest inference speed due to the limited computational resources and input sizes. Yet, it obtained the best AP values for five classes and the highest mAP value.

TABLE 7. Performances of different defect detection approaches in recent research.

ID	Defect dataset		Method	Performance						Year	Ref
	Defect type	Sample size		mAP_{50}	mAP_{70}	mAP_{75}	mAP	Recall	Speed		
1	4	3,000	Faster R-CNN	--	83%	--	--	--	9 FPS	2018	(Cheng and Wang 2018)
2	7	4,056	YOLOv3	85.37%	--	--	--	--	**33** FPS	2020	(Yin et al. 2020)
3	5	10,000	Faster R-CNN with SRPN	--	72.5%	--	--	89.5%	6 FPS	2021	(Li, Xie, et al. 2021)
4	3	3,600	Faster R-CNN	77%	--	--	--	--	9 FPS	2021	(Wang, Kumar, and Cheng 2021)
5	**8**	**14,010**	Pipe-VFNet (proposed)	**95.7%**	--	80.4%	73.4%	--	12 FPS	2022	This study

TABLE 8. Performances of different object detection approaches based on the proposed dataset.

Method	DS	HC	VC	JF	JO	LP	PB	SD	All (mAP)	Speed (FPS)
Faster R-CNN (Ren et al. 2015)	0.823	0.476	0.638	0.787	0.718	0.779	**0.548**	0.521	0.661	10.2
YOLOv3 (Redmon and Farhadi 2018)	0.818	0.326	0.613	0.708	0.648	0.789	0.513	0.457	0.609	**24.1**
RetinaNet (Lin et al. 2017)	0.826	0.386	0.625	0.754	0.603	0.793	0.496	0.580	0.633	11.9
ATSS (Zhang, Chi, et al. 2020)	0.812	0.284	0.620	0.800	0.650	0.770	0.447	0.746	0.641	13.3
VFNet (Zhang et al. 2021)	0.821	0.449	**0.735**	**0.867**	0.739	0.726	0.532	0.563	0.679	12.9
Pipe-VFNet (proposed)	**0.881**	**0.569**	0.704	0.850	**0.768**	**0.794**	0.542	**0.766**	**0.734**	12.4

6 CONCLUSION

In this study, an automatic defect detection framework is proposed to classify and localize eight types of defects that are frequently encountered in underground sewer pipelines. An attention module is first introduced and adopted to improve the feature extraction ability in the detector's backbone. Next, a new feature fusion mechanism that can lighten the feature dilution problem is used in the neck. Finally, a loss function is presented and used in the proposed

framework for solving the IDP. Experimental results proved that the detection framework deal with the imbalanced training samples well due to the effective loss function. In addition, the proposed network could detect defects under varying conditions with the highest mAP of 73.4% on the dataset. However, the processing speed of this defect detection framework is about 12 FPS on a Tesla GPU, which cannot fulfill the requirement of real-time detection.

In the future, more concern should be paid to the development of the proposed framework on various portable devices that facilitate realistic defect inspections. Besides, the processing speed of the defect detection network should be boosted for real-time applications. Furthermore, more powerful and effective algorithms, such as dynamic classification (Rafiei and Adeli 2017), fast learning (Pereira et al. 2020), and dynamic ensemble learning (Alam, Siddique, and Adeli 2020), should be investigated in the subsequent research.

ACKNOWLEDGMENT

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and by a grant (20212020900150) from "Development and Demonstration of Technology for Customers Bigdata-based Energy Management in the Field of Heat Supply Chain" funded by Ministry of Trade, Industry and Energy of Korean government and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-00106).

REFERENCES

'2021 Report Card for America's Infrastructure - Wastewater'. 2021.

Alam, Kazi Md Rokibul, Nazmul Siddique, and Hojjat Adeli. 2020. 'A dynamic ensemble learning algorithm for neural networks', *Neural Computing and Applications*, 32: 8675-90.

Ban, Ming-Yang, Wei-Dong Tian, and Zhong-Qiu Zhao. 2020. "Real-Time Object Detection Based on Convolutional Block Attention Module." In *International Conference on Intelligent Computing*, 41-50. Springer.

Cai, Zhaowei, and Nuno Vasconcelos. 2019. 'Cascade r-cnn: High quality object detection and instance segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cao, Jiale, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. 2020. "D2det: Towards high quality object detection and instance segmentation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11485-94.

Cao, Junxu, Qi Chen, Jun Guo, and Ruichao Shi. 2020. 'Attention-guided context feature pyramid network for object detection', *arXiv preprint arXiv:2005.11475*.

Chen, Yuhan, Shangping Zhong, Kaizhi Chen, Shoulong Chen, and Song Zheng. 2019. "Automated detection of sewer pipe defects based on cost-sensitive convolutional neural network." In *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*, 8-17.

Cheng, Jack CP, and Mingzhu Wang. 2018. 'Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques', *Automation in Construction*, 95: 155-71.

Czimmermann, Tamás, Gastone Ciuti, Mario Milazzo, Marcello Chiurazzi, Stefano Roccella, Calogero Maria Oddo, and Paolo Dario. 2020. 'Visual-based defect detection and classification approaches for industrial applications—a survey', *Sensors*, 20: 1459.

Dang, L Minh, SeonJae Kyeong, Yanfen Li, Hanxiang Wang, Tan N Nguyen, and Hyeonjoon Moon. 2021. 'Deep learning-based sewer defect classification for highly imbalanced dataset', *Computers & Industrial Engineering*, 161: 107630.

Gao, Shanghua, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. 2019. 'Res2net: A new multi-scale backbone architecture', *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Haurum, Joakim Bruslund, Meysam Madadi, Sergio Escalera, and Thomas B Moeslund. 2022. "Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2806-17.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-78.

Hou, Qibin, Daquan Zhou, and Jiashi Feng. 2021. "Coordinate attention for efficient mobile network design." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13713-22.

Hu, Jie, Li Shen, and Gang Sun. 2018. "Squeeze-and-excitation networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132-41.

Khan, Muhammad Safeer, and Rajvardhan Patil. 2018. "Acoustic characterization of pvc sewer pipes for crack detection using frequency domain analysis." In *2018 IEEE International Smart Cities Conference (ISC2)*, 1-5. IEEE.

Langerman, David, Alex Johnson, Kyle Buettner, and Alan D George. 2020. "Beyond Floating-Point Ops: CNN Performance Prediction with Critical Datapath Length." In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 1-9. IEEE.

Lepot, Mathieu, Nikola Stanić, and François HLR Clemens. 2017. 'A technology for sewer pipe inspection (Part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification', *Automation in Construction*, 73: 1-11.

Li, Dawei, Yida Li, Qian Xie, Yuxiang Wu, Zhenghao Yu, and Jun Wang. 2021. 'Tiny Defect Detection in High-Resolution Aero-Engine Blade Images via a Coarse-to-Fine Framework', *IEEE Transactions on Instrumentation and Measurement*, 70: 1-12.

Li, Dawei, Qian Xie, Zhenghao Yu, Qiaoyun Wu, Jun Zhou, and Jun Wang. 2021. 'Sewer pipe defect detection via deep learning with local and global feature fusion', *Automation in Construction*, 129: 103823.

Li, Jiajun, Qian Wang, Jun Ma, and Jingjing Guo. 2022. 'Multi‐defect segmentation from façade images using balanced copy‐paste method', *Computer‐Aided Civil and Infrastructure Engineering*, 37: 1434-49.

Li, Xiang, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 2020. 'Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection', *arXiv preprint arXiv:2006.04388*.

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. "Focal loss for dense object detection." In *Proceedings of the IEEE international conference on computer vision*, 2980-88.

Liu, Shu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. "Path aggregation network for instance segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759-68.

Lu, Zhenyu, Yanzhong Bai, Yi Chen, Chunqiu Su, Shanshan Lu, Tianming Zhan, Xunning Hong, and Shuihua Wang. 2020. 'The classification of gliomas based on a pyramid dilated convolution resnet model', *Pattern Recognition Letters*, 133: 173-79.

Lu, Zhou, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. 2017. "The expressive power of neural networks: A view from the width." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6232-40.

Luo, Yihao, Xiang Cao, Juntao Zhang, Jingjuan Guo, Haibo Shen, Tianjiang Wang, and Qi Feng. 2021. 'CE-FPN: Enhancing Channel Information for Object Detection', *arXiv preprint arXiv:2103.10643*.

Molchanov, Pavlo, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. 'Pruning convolutional neural networks for resource efficient inference', *arXiv preprint arXiv:1611.06440*.

Mostafa, Kareem, and Tarek Hegazy. 2021. 'Review of image-based analysis and applications in construction', *Automation in Construction*, 122: 103516.

Pereira, Danilo R, Marco Antonio Piteri, André N Souza, João Paulo Papa, and Hojjat Adeli. 2020. 'FEMa: A finite element machine for fast learning', *Neural Computing and Applications*, 32: 6393-404.

Rafiei, Mohammad Hossein, and Hojjat Adeli. 2017. 'A new neural dynamic classification algorithm', *IEEE transactions on neural networks*

and learning systems, 28: 3074-83.

Redmon, Joseph, and Ali Farhadi. 2018. 'Yolov3: An incremental improvement', *arXiv preprint arXiv:1804.02767*.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. 'Faster r-cnn: Towards real-time object detection with region proposal networks', *Advances in neural information processing systems*, 28: 91-99.

Wang, Hanxiang, Yanfen Li, L Minh Dang, Jaesung Ko, Dongil Han, and Hyeonjoon Moon. 2020. 'Smartphone-based bulky waste classification using convolutional neural networks', *Multimedia Tools and Applications*, 79: 29411-31.

Wang, Hanxiang, Yanfen Li, L Minh Dang, Sujin Lee, and Hyeonjoon Moon. 2021. 'Pixel-level tunnel crack segmentation using a weakly supervised annotation approach', *Computers in Industry*, 133: 103545.

Wang, Mingzhu, Srinath Shiv Kumar, and Jack CP Cheng. 2021. 'Automated sewer pipe defect tracking in CCTV videos based on defect detection and metric learning', *Automation in Construction*, 121: 103438.

Wang, Mingzhu, Han Luo, and Jack CP Cheng. 2021. 'Towards an automated condition assessment framework of underground sewer pipes based on closed-circuit television (CCTV) images', *Tunnelling and Underground Space Technology*, 110: 103840.

Weber, Michael, Michael Fürst, and J Marius Zöllner. 2020. "Automated focal loss for image based object detection." In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1423-29. IEEE.

Wu, Xiongwei, Doyen Sahoo, Daoxin Zhang, Jianke Zhu, and Steven CH Hoi. 2020. 'Single-shot bidirectional pyramid networks for high-quality object detection', *Neurocomputing*, 401: 1-9.

Wu, Yuxin, and Kaiming He. 2018. "Group normalization." In *Proceedings of the European conference on computer vision (ECCV)*, 3-19.

Xie, Qian, Dawei Li, Jinxuan Xu, Zhenghao Yu, and Jun Wang. 2019. 'Automatic detection and classification of sewer defects via hierarchical deep learning', *IEEE Transactions on Automation Science and Engineering*, 16: 1836-47.

Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. "Aggregated residual transformations for deep neural networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492-500.

Yin, Xianfei, Yuan Chen, Ahmed Bouferguene, Hamid Zaman, Mohamed Al-Hussein, and Luke Kurach. 2020. 'A deep learning-based framework for an automated defect detection system for sewer pipes', *Automation in Construction*, 109: 102967.

Yin, Xianfei, Tianxin Ma, Ahmed Bouferguene, and Mohamed Al-Hussein. 2021. 'Automation for sewer pipe assessment: CCTV video interpretation algorithm and sewer pipe video assessment (SPVA) system development', *Automation in Construction*, 125: 103622.

Zhang, Hang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, and R Manmatha. 2020. 'Resnest: Split-attention networks', *arXiv preprint arXiv:2004.08955*.

Zhang, Haoyang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. 2021. "Varifocalnet: An iou-aware dense object detector." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8514-23.

Zhang, Shifeng, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. 2020. "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9759-68.

Zhang, Youqi, and Weiwei Lin. 2022. 'Computer‐vision‐based differential remeshing for updating the geometry of finite element model', *Computer‐Aided Civil and Infrastructure Engineering*, 37: 185-203.

Zhou, Zhong, Junjie Zhang, and Chenjie Gong. 2022. 'Automatic detection method of tunnel lining multi‐defects via an enhanced You Only Look Once network', *Computer‐Aided Civil and Infrastructure Engineering*, 37: 762-80.

Zhu, Yousong, Chaoyang Zhao, Haiyun Guo, Jinqiao Wang, Xu Zhao, and Hanqing Lu. 2018. 'Attention couplenet: Fully convolutional attention coupling network for object detection', *IEEE Transactions on Image Processing*, 28: 113-26.