Journal of
Hydroinformatics

Journal of Hydroinformatics Vol 24 No 2, 406 doi: 10.2166/hydro.2022.132

Comparison of classic object-detection techniques for automated sewer defect detection

Qianqian Zhou[a,*], Zuxiang Situ[a], Shuai Teng[a], Weifeng Chen[a], Gongfa Chen[a] and Jiongheng Su[b]

[a] School of Civil and Transportation Engineering, Guangdong University of Technology, No. 100 Waihuan Xi Road, Guangzhou 510006, China
[b] Urban Development Research Center, Guangdong Urban & Rural Planning and Design Institute, No. 483 Nanzhou Road, Guangzhou 510290, China
*Corresponding author. E-mail: qiaz@foxmail.com

QZ, 0000-0003-3692-9498; ZS, 0000-0003-0503-0594

ABSTRACT

Sewer systems play a key role in cities to ensure public assets and safety. Timely detection of defects can effectively alleviate system deterioration. Conventional manual inspection is labor-intensive, error-prone and expensive. Object detection is a powerful deep learning technique that can complement and/or replace conventional inspection, especially in complex environments. This study compares two classic object-detection methods, namely faster region-based convolutional neural network (R-CNN) and You Only Look Once (YOLO), for the detection and localization of five types of sewer defects. Model performances are evaluated based on their detection accuracy and processing speed under parameterization impacts of dataset size and training parameters. Results show that faster R-CNN achieved higher prediction accuracy. Training dataset size and maximum number of epochs (MaxE) had dominant impacts on model performances of faster R-CNN and YOLO, respectively. The processing speed increased along with the increasing training data for faster R-CNN, but did not vary significantly for YOLO. The models' abilities to detect disjoint and residential wall were highest, whereas crack and tree root were more difficult to detect. The results help to better understand the strengths and weaknesses of the classic methods and provide a useful user guidance for practical applications in automated sewer defect detection.

Key words: deep learning, faster R-CNN, object detection, sewer defect detection, YOLO

HIGHLIGHTS

- A deep learning technique for automated detection of multiple types of sewer defects.
- Compared the performances of two types of classic object-detection models.
- Evaluated model parameterization impacts and identification of key factors.

GRAPHICAL ABSTRACT

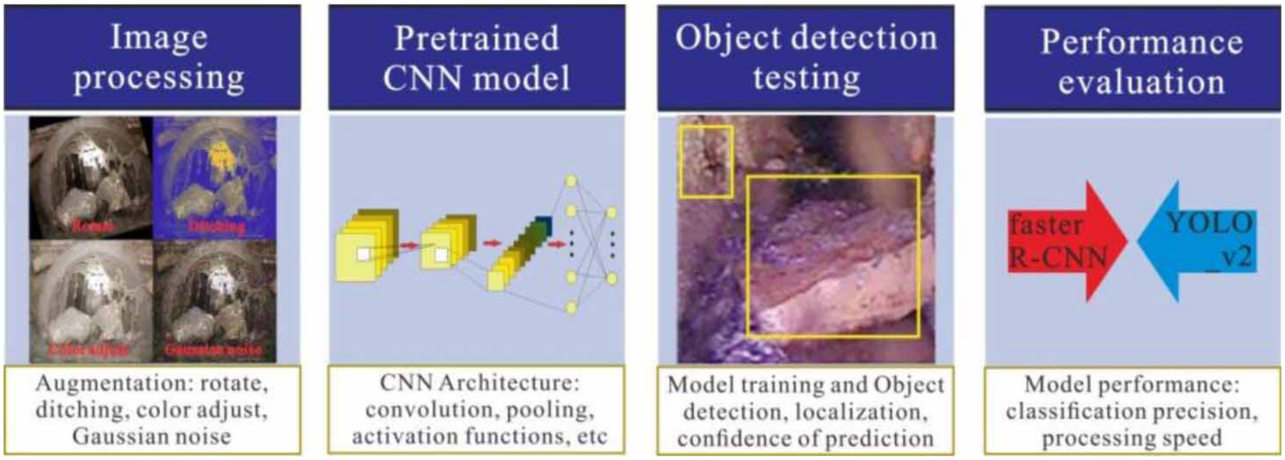
Image processing
Rotate
Ditching
Color adjust
Gaussian noise
Augmentation: rotate, ditching, color adjust, Gaussian noise
Pretrained CNN model
CNN Architecture: convolution, pooling, activation functions, etc
Object detection testing
Model training and Object detection, localization, confidence of prediction
Performance evaluation
faster R-CNN
YOLO _v2
Model performance: classification precision, processing speed

INTRODUCTION

Sewer systems play a key role in cities to ensure public assets and safety by transporting excess water from urban areas to receiving water or treatment facilities (Butler & Davies 2010; Xie *et al.* 2017; Cheng & Wang 2018; Zhou *et al.* 2019). Sewers without proper operation and maintenance are often associated with structural and/or functional failures and regular inspections are necessary to ensure system performance and reliability (Tafuri & Selvakumar 2002; Stanić *et al.* 2014). Given the large budget and resources on sewer systems, inspection of sewer defects at an early stage is especially beneficial, so that the system deteriorations can be detected, fixed or avoided in a timely manner by taking appropriate measures (Cheng & Wang 2018; Yin *et al.* 2020). Nevertheless, achieving rapid and accurate detection and classification of sewer defects remains a very challenging task, given the huge number of pipelines and their complex and varying associated conditions (Kumar *et al.* 2018; Hassan *et al.* 2019; Meijer *et al.* 2019).

Nowadays, the closed-circuit television (CCTV) inspection systems are widely applied to examine and record sewer conditions. The tool is especially necessary under unfavorable circumstances, such as high pressure, toxic and unsanitary environment (Wirahadikusumah *et al.* 2001; Hassan *et al.* 2019; Meijer *et al.* 2019; Yin *et al.* 2020). The CCTV videos are further sent to off-site technologists for an assessment of sewer defects based on their expert knowledge and personal experience. Such a manual assessment process is time-consuming, labor-intensive and thus associated with inconsistencies and uncertainties due to subjective assessment (Dirksen *et al.* 2013; Meijer *et al.* 2019; Xie *et al.* 2019; Yin *et al.* 2020). There is a pressing need to develop automated sewer defect detection methods, so that the large amount of time and resources spent on the conventional manual assessment can be saved. This can also greatly benefit the sewer maintenance and management in the long term (Cheng & Wang 2018; Jiang *et al.* 2019; Kumar & Abraham 2019; Xie *et al.* 2019).

The deep learning techniques that have been developed rapidly in the past decades are the pathway toward the automated methods (Cheng & Wang 2018; Xie *et al.* 2019; Yin *et al.* 2020). The technique has been successfully applied in different fields including water resources (Roushangar & Alizadeh 2018; Roushangar *et al.* 2021). One of the most representative and popular algorithms, namely the convolutional neural networks (CNNs), has been increasingly used in the field of pipe defect classification (Goodfellow *et al.* 2016; Gu *et al.* 2018). Compared to conventional image processing methods, the CNN undertakes both supervised and unsupervised learning and minimizes the complex feature extraction and training processes, which is thus superior to the conventional methods in terms of both processing speed and detection accuracy (Hassan *et al.* 2019; Kumar & Abraham 2019; Meijer *et al.* 2019; Xie *et al.* 2019). Despite the advantages, the image classification-based CNN can only detect a single type of defect at a time for image/defect classification and thus does not perform well in real-life inspection as complex sewer conditions (e.g., coexistence of multiple types of sewer defects) often exist (Krizhevsky *et al.* 2017).

In recent years, technologies that can detect, classify and localize multiple defects simultaneously, namely object detection, have gained widespread attention (Cheng & Wang 2018; Jiang *et al.* 2019; Kumar & Abraham 2019; Yin *et al.* 2020). There are several object-detection algorithms, which are the extensions of CNNs, including the region-based CNN (R-CNN) (Girshick *et al.* 2014), fast R-CNN (Girshick 2015) and faster R-CNN (Ren *et al.* 2017). The main improvements of these algorithms are the use of innovative methods to find the region of interest (ROI), such as selective search of R-CNN, convolutional feature map of the fast R-CNN and region proposal network (RPN) of the faster R-CNN (Girshick *et al.* 2014; Girshick 2015; Ren *et al.* 2017). For example, Cheng & Wang (2018) adopted a faster R-CNN to automate the detection and localization of tree root, deposit, infiltration and crack inside sewer pipes. Their results showed that faster R-CNN has high precision and recall value and enables an accurate detection of sewer defects. Nevertheless, these algorithms were reported to be slow due to their two-staged processing approach.

On the other hand, the one-stage detection method, such as the most well-known YOLO by Redmon & Farhadi (2017), was reported to be faster than the faster R-CNN method due to the removal of the region proposal mechanism. YOLO uses only one CNN (the faster R-CNN including one RPN and one CNN) to directly predict the classifications and locations of different objects in an image. Thus, in theory, the YOLO model has a higher calculation speed, but with some loss in accuracy as it was difficult to solve the problem of precise location and classification. Kumar & Abraham (2019) adopted a YOLO model to detect pipe fractures. Yin *et al.* (2020) applied a YOLO-based object detector for a real-time detection of six types of sewer defects.

Nevertheless, research comparing the performances of the two classic types of object-detection algorithms has been limited, especially for detecting multiple types of sewer defects in a consistent framework. This study presents the object-detection-based deep learning technology for accurate and efficient detection of multiple types of sewer defects as a complement and/or replacement for conventional manual inspection. The faster R-CNN and YOLO_v2 models were compared for automated sewer defect detection. Model performances were evaluated in terms of detection accuracy and processing speed. Five types of sewer defects, namely crack (CR), disjoint (DJ), obstacle (OB), residential wall (RW) and tree root (TR) were investigated. The results obtained will provide more insights into the performance (e.g., strengths and weaknesses) of the classic object-detection techniques for sewer defect detection and proper guidance/references on the application and selection of an appropriate method for practical applications.

MATERIALS AND METHODS

The overall workflow of this study (Figure 1) consists of (1) sewer defect image processing, including data augmentation and annotation techniques, (2) a CNN model, pertained as the feature extractor for object-detection methods, (3) object-detection model training and testing: faster R-CNN vs. YOLO_v2 and (4) performance evaluation and comparison. Details are explained in the following sections.

Image processing

The investigated sewer images include five types of defects, namely CR, DJ, OB, RW and TR. The five types of defects were selected because they are the most commonly encountered sewer defects in southern China (Lin 2014; Qi *et al.* 2017; Xiao *et al.* 2019). The original images were obtained via multiple sources of CCTV videos under various pipe conditions. The images were further examined and assigned with their own defect labels by sewer experts. All images were rescaled to a

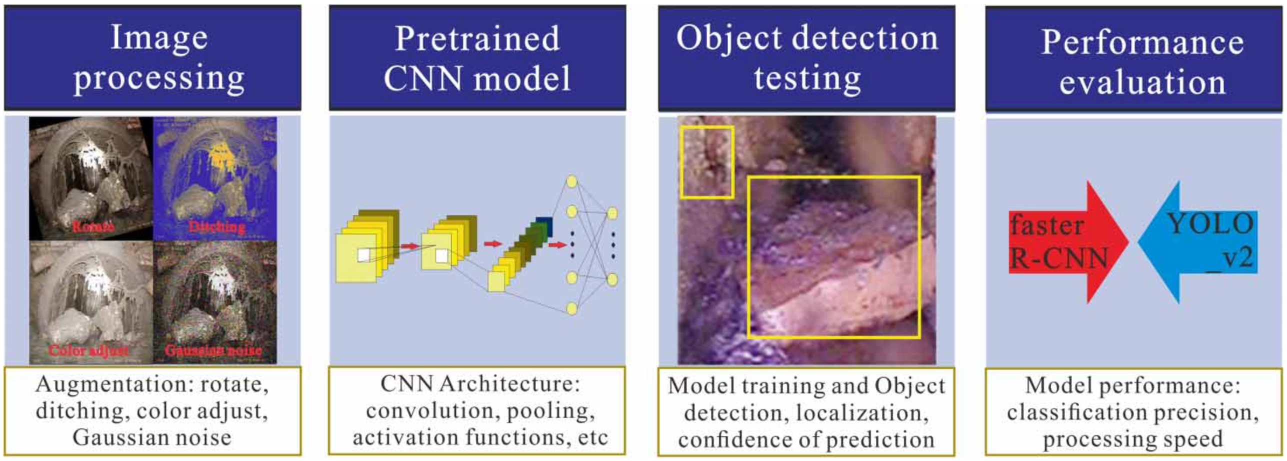
Image processing
Rotate
Ditching
Color adjust
Gaussian noise
Augmentation: rotate, ditching, color adjust, Gaussian noise
Pretrained CNN model
CNN Architecture: convolution, pooling, activation functions, etc
Object detection testing
Model training and Object detection, localization, confidence of prediction
Performance evaluation
faster R-CNN
YOLO _v2
Model performance: classification precision, processing speed

Figure 1 | Illustration of workflow of the study.

uniform 256 × 256 pixel resolution based on technical details suggested in the previous literature (Cha *et al.* 2017; Xu *et al.* 2020) to retain the image features and meanwhile obtain high computational efficiency, despite the fact that images with higher resolution can provide more information for model training (Cheng & Wang 2018).

To improve network detection accuracy and prevent overfitting phenomenon, the data augmentation techniques were employed to increase the size and variety of the model dataset (Krizhevsky *et al.* 2017; Kumar *et al.* 2018; Douarre *et al.* 2019; Zhang *et al.* 2019). Data augmentation is an important technique, especially for limited data conditions, to generate new and representative data and thus enhance the quantity and quality of data. Consequently, the technique can significantly improve the performance of the neural network (Li *et al.* 2020b; Rodriguez-Lozano *et al.* 2020; Xu *et al.* 2020). In summary, the geometric transformations (including rotation, mirror and translation) and color transformation operations (including ditching, color adjust, noises with Gaussian, Salt and pepper, and Poisson processing) were applied (Figure 2). Details on the applied data augmentation are shown in Table 1, and it can be seen that there are in total 11 operations adopted. The original number of defect images was 610 and was enriched to 7,320 for model training and validation. During the data augmentation, we kept the image format (.tif format) and resolution of each image unchanged. When using the color transformation, the brightness and definition of image may be changed, but inputting images with different qualities can ensure higher robustness of the model. Finally, the 'Image Labeler' toolbox in MATLAB was employed to label the

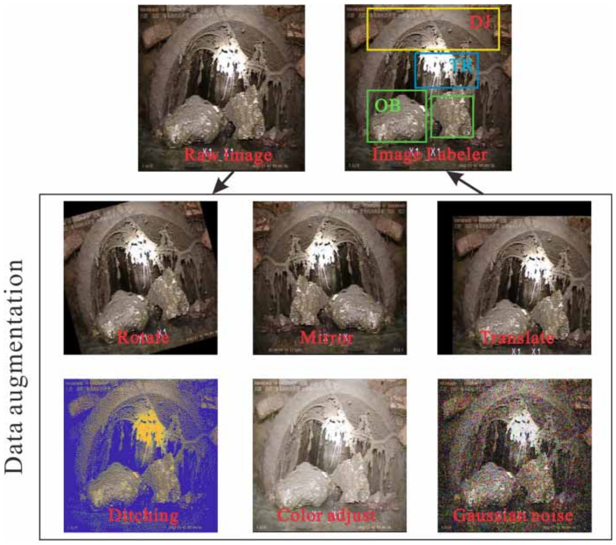

Figure 2 | Sample images of data augmentation and class labeler.

Table 1 | Data augmentation operations adopted in the study

General type	Subtype	Description
Geometric transformations	Rotation	Rotate the image by 15° in a counterclockwise. Rotate the image by 15° in a clockwise.
	Mirror	Mirror the input image horizontally.
	Translation	Translate the input image by the [25, 25] vector specified in translation. Translate the input image by the [−25, − 25] vector specified in translation.
Color transformations	Ditching	Create an indexed image approximation of the input image by dithering the colors in the Parula colormap.
	Color adjustment	Adjust the image grayscale and increase brightness with nonlinear mapping. Adjust the image grayscale and reduce brightness in a nonlinear mapping.
	Adding noise	Add Gaussian white noise with the mean value of 0 and variance selected randomly from the range [0.05, 0.1]. Add Salt and pepper noise with a noise density selected randomly from the range [0.1, 0.3]. Add Poisson noise.

rectangular ROI (i.e., ground-truth bounding boxes) and associated class ID (namely, CR, DJ, OB, RW and TR) for object detection of the sewer defects.

CNN-based feature extractor

A pretrained CNN model (Zhou *et al.* 2021) is used as the feature extractor for both faster R-CNN and YOLO_v2 (Figure 3). There are several essential layers with different functions during the feature extraction: (1) convolutional layer (CONV), to extract local features from input images. The main operation of convolution is to obtain the weighted sums of local regions for feature maps by multiplying the elements in the convolutional kernel with the elements in the input data (Cha *et al.* 2017; Gu *et al.* 2018; Xie *et al.* 2019). (2) Activation layer with Rectified Linear Units (ReLUs), which adds non-linearity between different layers to allow the model to converge faster and enhance the computing capability (Nair & Hinton 2010; Krizhevsky *et al.* 2017). (3) Max pooling layer (MaxPOOLing), to reduce the dimensions of feature maps by down sampling. This process is to take the maximum values of different local regions at the prior layer/input layer (Scherer *et al.* 2010; Xie *et al.* 2019; Teng *et al.* 2020). (4) Fully connected layer, to compute the class probability by generating final nonlinear combinations of features (Hassan *et al.* 2019; Meijer *et al.* 2019). Meanwhile, the softmax function is used to calculate the output probability scores for the predicted classes. The detailed architecture of the pretrained CNN model is shown in Figure 3, which consists of one input layer, three convolution layers, two max pooling layers, three activation layers (i.e., kernel function), one fully-connected layer and one output layer. The softmax layer was set behind the fully connected layer and before the output layer.

Object-detection models

Faster R-CNN

As shown in Figure 4(a), there are three main components in the faster R-CNN model. First, the abovementioned CNN model is utilized in the first step to extract features from input images. Second, based on the feature maps, RPN generates various region proposals with different scales and aspect ratios using anchors. Third, the region proposals are input to the ROI layer and undergo the last convolution process (Equation (1)) and ReLU function (Equation (2)) for further refining in the faster R-CNN detector. Finally, the extracted features are fed into two fully connected layers, i.e., a softmax layer producing probability scores (Equation (3)) for the detected sewer defects and a bounding box regression layer (Equations (4)–(7)) for producing the relative location coordinates. In doing so, multiple types of sewer defects can be detected and localized by bounding boxes with a predicted accuracy.

$$f(i) = \sum_{n=1}^{v_k} S(i+n)K(n) \tag{1}$$

$$f(x) = \max(0, x) \tag{2}$$

where S and K are the input (usually a vector or matrix) and convolution kernel, respectively, $f(i)$ is feature map (output) of the ith input (S) and convolutional kernel (K) and n is the maximum index (number of weights) of the K. i is the number of feature values contained in the feature map, and k is the number of convolutional kernels. In Equation (2), the x represents each value of input (S).

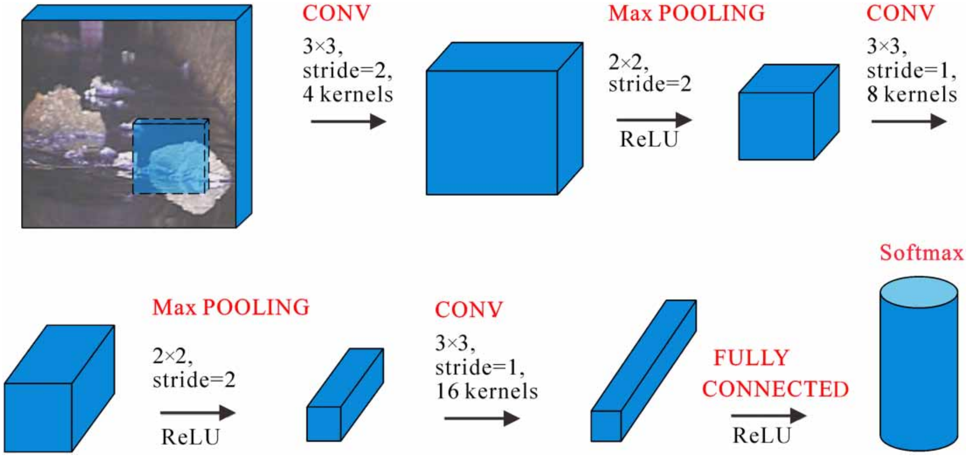

Figure 3 | CNN-based feature extractor network architecture.

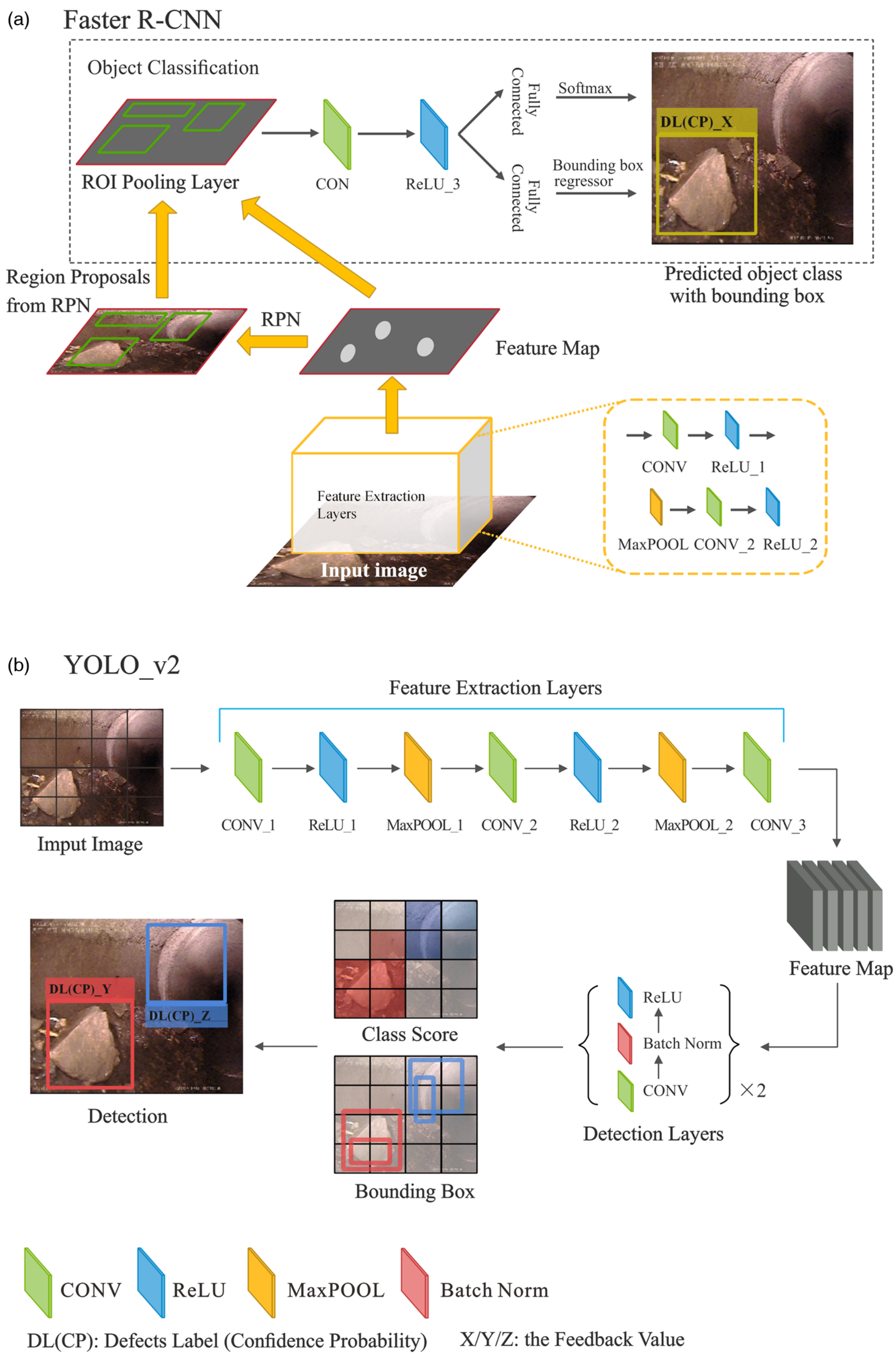

Figure 4 | Architecture of the applied (a) faster R-CNN model and (b) YOLO_v2 model.

The prediction probability of class *j* is defined in the following equation:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{3}$$

where p_i is the probability that the input sample (z_i) belongs to class i and N is the number of classes. In the bounding box regression layer, assuming that the central coordinates of the bounding box are (A_x, A_y), the length and height are A_w and A_h, respectively, and the value of the ground truth is [G_x, G_y, G_w, G_h]; therefore, the offset of relevant parameters is:

$$d_x(A) = \frac{(G_x - A_x)}{A_w} \tag{4}$$

$$d_y(A) = \frac{(G_y - A_y)}{A_h} \tag{5}$$

$$d_w(A) = \log\left(\frac{G_w}{A_w}\right) \tag{6}$$

$$d_h(A) = \log\left(\frac{G_h}{A_h}\right) \tag{7}$$

where [$d_x(A)$, $d_y(A)$, $d_w(A)$, $d_x(A)$] is the offset of the bounding box from the ground truth.

Several inputs were further specified to parameterize the faster R-CNN network: (1) network input size, (2) feature extraction network and (3) training options. In this study, we specified the network input size of [256 256 3], i.e., 256 × 256 pixel resolution with three channels (i.e., RGB image), to ensure a reasonable computing time and thus all images were resized in the image processing prior to the training step. As for the feature extraction layer, the last CONV of the pretrained CNN model was selected based on empirical analysis. The design of network layer was primarily based on the official manuals of the MATLAB and our previous research results (Teng *et al.* 2021; Lin *et al.* 2022). The previous work has explained the functions of all layers in detail and confirmed that these layers have excellent performance for defect image detection. Also, these layers can obtain the most abundant defect features. Furthermore, option parameters such as initial learning rate, solver for training network, maximum number of epochs and size of mini-batch can be defined to specify the network training options of the model. Finally, the faster R-CNN object detector was set and trained by calling the 'trainFasterRCNNObjectDetector' tool with the training dataset in the MATLAB.

YOLO_v2 model

YOLO_v2 treats object detection as regression through end-to-end training (Redmon & Farhadi 2017; Yin *et al.* 2020). As shown in Figure 4(b) the major difference between the two models is that instead of using region proposals for feature extraction, YOLO_v2 adopted single-stage detection and drives the input image into pre-defined grid cells for classification. The model runs the pretrained CNN directly on input images as the feature extractor in the detection network. Unlike the faster R-CNN, the YOLO can more directly predict both class probability and bounding box locations/coordinates (Equations (8)–(11)) simultaneously for each object on the input image (Kumar & Abraham 2019). Specifically, for each grid cell, the model evaluates the class probability and predicts the classes of objects using bounding box. In the output, there is a confidence value associated with each bounding box, which indicates how precisely the box fits the object.

$$d_x(A)' = G_x - A_x \tag{8}$$

$$d_y(A)' = G_y - A_y \tag{9}$$

$$d_w(A)' = \log\left(\frac{G_w}{A_w}\right) \tag{10}$$

$$d_h(A)' = \log\left(\frac{G_h}{A_h}\right) \tag{11}$$

where [$d_x(A)'$, $d_y(A)'$, $d_w(A)'$, $d_h(A)'$] is the offset of the bounding box from the ground truth. In particular, the calculation method of the offset in the x and y directions of the YOLO_v2 is different from that of the faster R-CNN. Regarding the model parameterization, the settings on the network input size, the pretrained CNN model and feature extraction network were the same as the ones of the faster R-CNN model. As for the anchor boxes, the MATLAB function 'estimate AnchorBoxes' was used to generate optimal anchor boxes based on the size of objects in the training dataset.

To create the YOLO_v2 object-detection network, the network input size, number of object classes, anchor boxes, base network and feature extraction layer need to be defined in the 'yolov2Layers' function and returned as a LayerGraph object (in the MATLAB platform). Similarly, the training options such as initial learning rate, solver for training network and maximum number of epochs were defined. Finally, with inputs of the training dataset, LayerGraph object and option setting, the YOLO_v2 detector was established by using the 'trainYOLOv2ObjectDetector' function. In addition, the training parameters for both faster R-CNN and YOLO_v2 included: (1) optimizer: SGDM; (2) mini-batch-size: 8 and (3) learning rate: 0.001. The training platform was performed on a computer with NVIDIA GTX GeForce 1650 GPU, Intel Core i7-4790@3.60 GHz CPU, Windows 10.

Experimental setup and performance evaluation

Experimental setup

The effects of three main parameters on model performance were investigated: (1) the proportion of total number of images to be used for model training and validation (i.e., ND). In total, there were six values of ND tested, which are 0.1, 0.2, 0.4, 0.6, 0.8 and 1, respectively. For example, if a value of 0.8 was adopted (i.e., ND = 0.8), then 80% of images in the total dataset were used for model training and validation. (2) The proportion of ND, namely Tp, describes the number of images for model training. The tested Tp values include 0.5, 0.6, 0.7, 0.8 and 0.9. Note that once Tp is assigned, the images were randomly divided into two independent datasets for training and validation, respectively. For example, when ND = 0.8 and Tp = 0.9, then 72% (80% × 90%) of total images were employed for model training. Note that the first two parameters are used to test the impacts of different partitions/percentages of images on model performance. (3) Maximum number of epoch (MaxE) – an epoch is the full pass of the training algorithm over the entire training set. We tested four types of MaxE in this study, namely 50, 100, 200 and 300.

To summarize, the first two parameters mainly define the size for different datasets and the third one is the main influencing factor in the model training process. The adopted values of the parameters were set according to the values suggested in the literature (Deng *et al.* 2020; Kumar *et al.* 2020; Li *et al.* 2020a, 2021). Theoretically, the higher the three parameters, the better detection performance anticipated. Both object-detection models were run for each combination of the three parameters (ND (namely, 0.1, 0.2, 0.4, 0.6, 0.8 and 1.0), Tp (namely, 0.5, 0.6, 0.7, 0.8 and 0.9) and MaxE (namely, 50, 100, 200 and 300)), and thus, there are, in total, $6 \times 5 \times 4 = 120$ groups of parameter combinations for model performance comparison. As the training and validation sets and MaxE differ in each simulation, the associated processing speed also varies.

Performance evaluation

Two types of commonly used performance metrics (Cheng & Wang 2018; Yin *et al.* 2020) are evaluated to measure the performance of models, namely the detection accuracy and processing speed. The detection accuracy contains three submetrics, including the precision (*P*, the ability of the detector to predict correct classifications, Equation (12)), recall (*R*, the ability of the detector to find all relevant objects, Equation (13)) and average precision (*AP*, incorporates the precision and recall metrics, Equation (14)). It is difficult to evaluate the detection effects based on *P* and *R* individually as they have an overall lack of information. Instead, AP is a comprehensive indicator of *P* and *R* and is thus commonly used to evaluate the network performance (Maeda *et al.* 2018; Deng *et al.* 2020; Zhang *et al.* 2020; Teng *et al.* 2021). The value range of AP is 0–1; the closer the AP value to 1, the more excellent the detection effect. As there are many classes of objects, the result of AP is a vector of scores of each object class. On the other hand, the computation cost refers to the time/processing speed required by the entire detection process to make predictions in this study. The unit of measurement is seconds.

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$AP = \sum_{k=1}^{N} P(k) \triangle R(k) \tag{14}$$

where *TP* (real example): positive samples are predicted to be positive samples. *FN* (false counter example): positive samples are predicted to be negative samples. *FP* (false-positive example): negative samples are predicted to be positive samples. *N* is the number of testing samples.

RESULTS AND DISCUSSION

The example images of the object-detection models are shown in Figure 5. Each bounding box is associated with two types of information, namely the predicted class label (in abbreviation) and the corresponding confidence level, respectively. It is seen that the models can identify and label multiple defects simultaneously, even in very complex conditions, such as in Figure 5(a), 5(c) and 5(d). Comparisons of model performances of the faster R-CNN and YOLO_v2 under parameterization impacts are illustrated by the compass plots in Figure 6. As indicated by the results, dataset size and training parameters can influence the performance of the defect detection models. The results of mean APs are shown in the inner-most black ring, where the APs were categorized into four classes (i.e., 0–25, 25–50, 50–75 and 75–100 quartiles) according to the respective values. Similarly, the corresponding processing time is categorized and illustrated in the green ring next to the mean AP results. The combinations of the influencing parameters investigated in this study are shown in the three outside rings (i.e., MaxE in yellow, Tp in blue and ND in red).

The model performance of faster R-CNN and YOLO_v2 based on combinations of the three parameters is shown in the compass plots in Figure 6. It summarizes the model performances under each combination of the investigated parameters (ND (namely, 0.1, 0.2, 0.4, 0.6, 0.8 and 1.0), Tp (namely, 0.5, 0.6, 0.7, 0.8 and 0.9) and MaxE (namely, 50, 100, 200 and 300)). Both compasses were thus sorted by the value of mean AP (i.e., the most inner circle) in descending order in the clockwise direction. Results show that high APs generally required longer processing time. Nevertheless, the parameterization impacts differed for the two models. For the faster R-CNN, ND had high influences on the performance of model performance and high APs were generally associated with high NDs. This indicates that the number of training images (i.e., size of training dataset) is important to enhance the faster R-CNN model prediction capability. For Tp and MaxE, there was a slight tendency that larger values contribute to higher APs. As for the YOLO_v2, results show that MaxE had the dominant impact on the model performance. The larger values of MaxE contributed to higher detection accuracy. However, the combinations of Tp and ND made no clear contributions to the mean AP values. This confirms that the faster R-CNN was more susceptible to relevant parameters than the YOLO_v2. The findings discussed are essential to identify important parameters in the model setting and guide further uses of the two types of object-detection models.

A statistical comparison of the model performance is shown in Figure 7. Specifically, when there was a small number of ND ($ND \leq 100$), YOLO_v2 achieved higher detection accuracy. Nevertheless, the mean APs for YOLO_v2 changed much less significantly in comparison to the faster R-CNN. With the increase of ND (i.e., dataset size), the faster R-CNN starts to outperform the YOLO_v2 in identifying sewer defects. The main reason can be that for the faster R-CNN, with more training data, the model has more input resources and thus learns the object features more precisely. Overall, the faster R-CNN

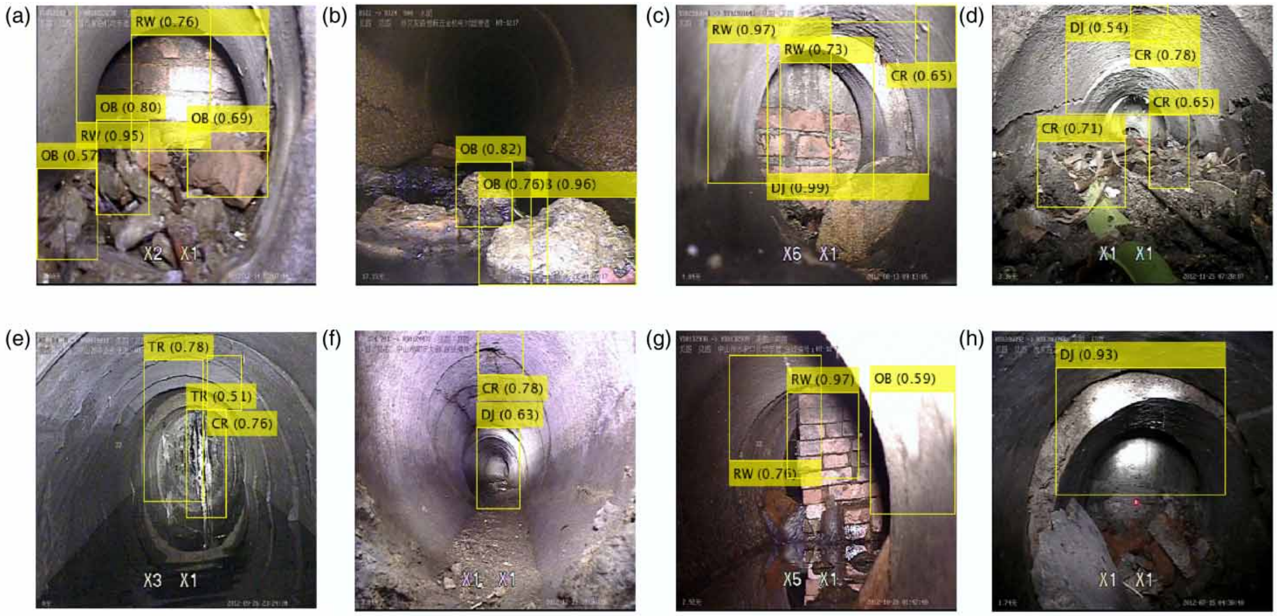

Figure 5 | Example images of the object-detection model performance.

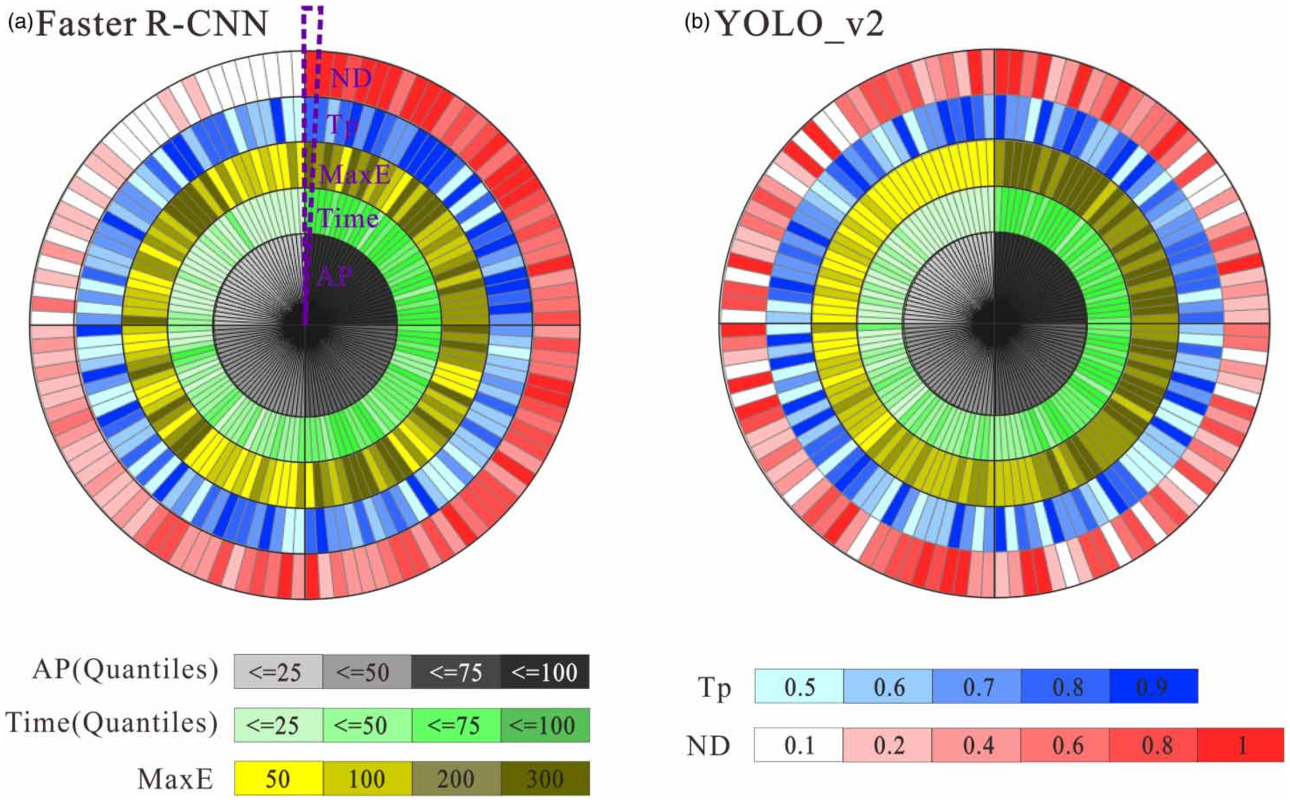

Figure 6 | AP and processing speed (time) of (a) faster R-CNN and (b) YOLO_v2 under parameterization impacts of MaxE, Tp and ND over the 120 runs. Each ring illustrates one type of parameter shown in the legend and each slot (highlighted by the dotted line) in the radial direction corresponds to a specific combination of investigated parameters.

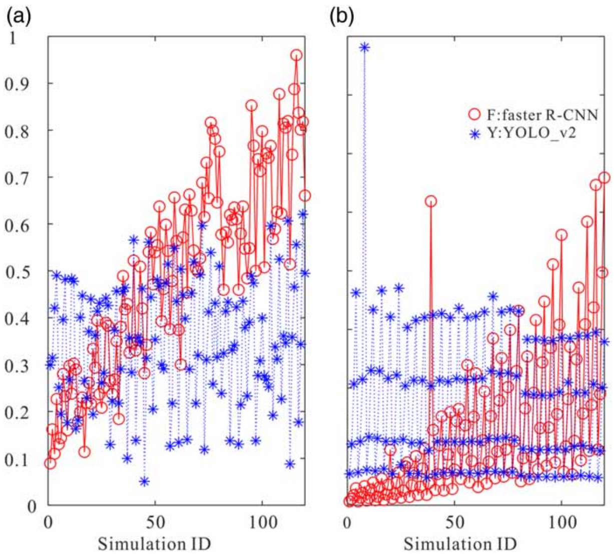

Figure 7 | Statistical summary of the relative (a) AP and (b) processing time of faster R-CNN and YOLO_v2 over the 120 runs.

has higher median mean AP in comparison to the YOLO_v2 in this study. As for processing speed, there was less time required by the faster R-CNN with a small value of ND. But note that despite the processing time being less for the faster R-CNN, the achieved relative AP was also lower. The computation cost increases along with the increasing ND. Interestingly, it was found that the processing time for the YOLO_v2 did not vary significantly over the entire simulation. Therefore, there was a trade-off between the detection accuracy and computation cost among the two types of models.

The result of AP in detecting each type of defect is shown in Figure 8. The results show that the faster R-CNN or YOLO_v2 models have different detection capabilities for different types of defects and thus the achieved detection accuracy differed for the investigated defects. The AP of disjoint was highest (higher AP means better detection effect), followed by ones of the residential wall and obstacle. The crack and tree root have much lower APs than other types of defects. This means the crack and tree root were difficult to be predicted by both current object-detection models. The reason may be attributed to the complexity in detecting the two types of defects, which have less distinct features to be identified in comparison to other defects. Second, there were often multiple simultaneous occurrences of the crack and tree root in one image and the locations of the two types of defects are more scattered. This makes it difficult to detect all of them at once and the recall rates are thus lower. Finally, we also examine the distribution of the five sewer defects in the training datasets. It is shown that in this study, the low APs are less likely to be attributed to the sample size of the two defects as their distribution proportions (the pie chart) did not differ greatly to other defects. The ratios of the five defects were kept as similar as possible to ensure the comparability of the model results.

CONCLUSIONS AND FUTURE WORK

This paper compares the performances of two types of classic object-detection models, namely faster R-CNN and YOLO_v2 in terms of their prediction accuracy and processing speed. We evaluated the model capabilities in detecting and localizing five types of sewer defects under parameterization impacts of dataset size and model parameter (i.e., maximum number of epoch). Results show that on the whole, faster R-CNN achieved a higher prediction accuracy according to the median values. Regarding the parameterization, the size of the training dataset has essential impacts on the model performance of the faster R-CNN. With more training data, the model learns the object feature more precisely. As for the YOLO_v2, the MaxE has a dominant contribution to its prediction accuracy. As for processing speed, the computation cost increases along with the increasing training data for the faster R-CNN. Nevertheless, the processing time for the YOLO_v2 did not vary significantly over the entire simulation. It is thus addressed that there is a trade-off between the detection accuracy

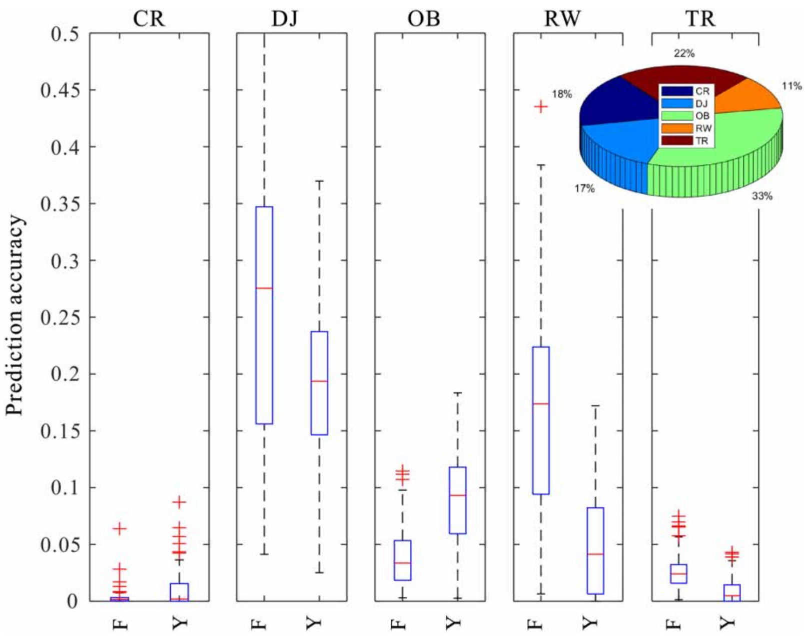

Figure 8 | Individual prediction accuracy of the five types of sewer defects achieved by the faster R-CNN (F) and YOLO_v2 (Y), respectively. The box edges illustrate the 25th and 75th percentiles of the data, the red line in the center of the box represents the median values and the whiskers mark the 5th and 95th percentiles. The distribution proportions of the five sewer defects in the datasets are illustrated in the pie chart. Please refer to the online version of this paper to see this figure in colour: http://dx.doi.org/10.2166/hydro.2022.132.

and computation cost among the two types of models. Furthermore, the models' abilities to detect disjoint and residential wall are highest with both object-detection models, whereas crack and tree root are more difficult to detect.

We acknowledge limitations of the current study for future work to better improve the model performance. The detection accuracies of both models are expected to be improved, which can be achieved by further research from four perspectives. The network structure and parameter of both models can be optimized to benefit the feature extraction and detection performance. The color, shape, brightness and contrast of the defect images can influence the model accuracy. We will look into more data augmentation techniques (e.g., generative adversarial network (GAN) generates a large number of virtual images with similar characteristics to real-world images) and combination strategies. Also, more original defect images are needed to enlarge the size of the training dataset to cover as many features of the multiple types of defects as possible. Meanwhile, improved label techniques are suggested. Other types of object-detection methods and/or transfer learning techniques are considered to improve the accuracy, such as trying SSD (Single Shot MultiBox Detector) algorithm, especially for multi-scale defects. What's more, one limitation of object detection is that the method cannot provide further detailed information on the geometric properties of sewer defects (e.g., shape, area and boundary), and future work will look into the application of the semantic segmentation technique that can provide a pixel-level segmentation of sewer defects for a more accurate description.

Despite the limitations, this study compares the performances of the two types of object-detection methods based on comparable input dataset and model settings. It is shown that both models have the capability to detect and localize multiple types of defects for sewer pipelines. This has great potential to complement the conventional labor-intensive manual sewer inspection. The results provide references for other studies to better understand the relative strengths and weaknesses of the two types of object-detection methods. The findings help to indicate the important influencing factors of the two models and provide insights for guiding further uses of the models. Equally important, the trade-off between detection accuracy and computation cost revealed for both models can be used as further references for practical applications.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant No. 51809049), the Science and Technology Program of Guangzhou, China (Grant No. 201804010406) and the National College Student Innovation and Entrepreneurship Training Program Project (Grant No. 202111845038).

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

REFERENCES

Butler, D. & Davies, J. 2010 *Urban Drainage*, 3rd edn. CRC Press, London. ISBN:0415455251.

Cha, Y. J., Choi, W. & Buyukozturk, O. 2017 Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering* **32** (5), 361–378.

Cheng, J. C. P. & Wang, M. 2018 Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Automation in Construction* **95**, 155–171.

Deng, J. H., Lu, Y. & Lee, V. C. S. 2020 Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering* **35** (4), 373–388.

Dirksen, J., Clemens, F. H. L. R., Korving, H., Cherqui, F., Le Gauffre, P., Ertl, T., Plihal, H., Müller, K. & Snaterse, C. T. M. 2013 The consistency of visual sewer inspection data. *Structure and Infrastructure Engineering* **9** (3), 214–228.

Douarre, C., Crispim-Junior, C. F., Gelibert, A., Tougne, L. & Rousseau, D. 2019 Novel data augmentation strategies to boost supervised segmentation of plant disease. *Computers and Electronics in Agriculture* **165**, 104967.

Girshick, R. 2015 Fast R-CNN. In: *IEEE International Conference on Computer Vision* (R. Bajcsy, G. Hager & Y. Ma, eds.). IEEE, Santiago, Chile, pp. 1440–1448.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. 2014 Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (S. Dickinson, D. Metaxas & M. Turk, eds.). IEEE, Columbus, OH. pp. 580–587.

Goodfellow, I., Bengio, Y. & Courville, A. 2016 *Deep Learning*. MIT Press, Cambridge, MA.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Chen, T. 2018 Recent advances in convolutional neural networks. *Pattern Recognition* **77**, 354–377.

Hassan, S. I., Dang, L. M., Mehmood, I., Im, S., Choi, C., Kang, J., Park, Y.-S. & Moon, H. 2019 Underground sewer pipe condition assessment based on convolutional neural networks. *Automation in Construction* **106**, 102849.

Jiang, J. C., Liu, J. Z., Cheng, C. X., Huang, J. Z. & Xue, A. K. 2019 Automatic estimation of urban waterlogging depths from video images based on ubiquitous reference objects. *Remote Sensing* **11** (5), 10.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2017 Imagenet classification with deep convolutional neural networks. *Communications of the ACMm* **60** (6), 84–90.

Kumar, S. S. & Abraham, D. M. 2019 A deep learning based automated structural defect detection system for sewer pipelines. In: *ASCE International Conference on Computing in Civil Engineering (i3CE)*. American Society of Civil Engineers, Georgia Institute of Technology, Atlanta, Georgia, pp. 226–233.

Kumar, S. S., Abraham, D. M., Jahanshahi, M. R., Iseley, T. & Starr, J. 2018 Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. *Automation in Construction* **91**, 273–283.

Kumar, S. S., Wang, M. Z., Abraham, D. M., Jahanshahi, M. R., Iseley, T. & Cheng, J. C. P. 2020 Deep learning-based automated detection of sewer defects in CCTV videos. *Journal of Computing in Civil Engineering* **34** (1), 13.

Li, B. X., Wang, K. C. P., Zhang, A., Yang, E. H. & Wang, G. L. 2020a Automatic classification of pavement crack using deep convolutional neural network. *International Journal of Pavement Engineering* **21** (4), 457–463.

Li, G., Ren, X. L., Qiao, W. T., Ma, B. & Li, Y. 2020b Automatic bridge crack identification from concrete surface using ResNeXt with postprocessing. *Structural Control & Health Monitoring* **27** (11), 20.

Li, D. W., Xie, Q., Yu, Z. H., Wu, Q. Y., Zhou, J. & Wang, J. 2021 Sewer pipe defect detection via deep learning with local and global feature fusion. *Automation in Construction* **129**, 13.

Lin, M. 2014 Health inspection and analysis of sewer system in an area of Fuzhou City. *China Water & Waste Water* **30** (9), 96–98.

Lin, M., Teng, S., Chen, G., Lv, J. & Hao, Z. 2022 Optimal CNN-based semantic segmentation model of cutting slope images. *Frontiers of Structural and Civil Engineering*. (in press). doi:10.1007/s11709-021-0797-6.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T. & Omata, H. 2018 Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering* **33** (12), 1127–1141.

Meijer, D., Scholten, L., Clemens, F. & Knobbe, A. 2019 A defect classification methodology for sewer image sets with convolutional neural networks. *Automation in Construction* **104**, 281–298.

Nair, V. & Hinton, G. E. 2010 Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, Haifa, Israel, pp. 807–814.

Qi, L., Zu, S. & Ma, J. 2017 CCTV inspection and thinking of a regional sewage networks in Zhuhai. *China Water & Waste Water* **33** (22), 135–138.

Redmon, J. & Farhadi, A. 2017 YOLO9000: better, faster, stronger. In: *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, pp. 6517–6525.

Ren, S., He, K., Girshick, R. & Sun, J. 2017 Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39** (6), 1137–1149.

Rodriguez-Lozano, F. J., Leon-Garcia, F., Gamez-Granados, J. C., Palomares, J. M. & Olivares, J. 2020 Benefits of ensemble models in road pavement cracking classification. *Computer-Aided Civil and Infrastructure Engineering* **35** (11), 1194–1208.

Roushangar, K. & Alizadeh, F. 2018 Scenario-based prediction of short-term river stage–discharge process using wavelet-EEMD-based relevance vector machine. *Journal of Hydroinformatics* **21** (1), 56–76.

Roushangar, K., Aghajani, N., Ghasempour, R. & Alizadeh, F. 2021 The potential of ensemble WT-EEMD-kernel extreme learning machine techniques for prediction suspended sediment concentration in successive points of a river. *Journal of Hydroinformatics* **23** (3), 655–670.

Scherer, D., Muller, A. & Behnke, S. 2010 Evaluation of pooling operations in convolutional architectures for object recognition. In *20th International Conference on Artificial Neural Networks*. Springer-Verlag, Berlin, Thessaloniki, Greece, pp. 92–101.

Stanić, N., Langeveld, J. G. & Clemens, F. H. L. R. 2014 HAZard and OPerability (HAZOP) analysis for identification of information requirements for sewer asset management. *Structure and Infrastructure Engineering* **10** (11), 1345–1356.

Tafuri, A. N. & Selvakumar, A. 2002 Wastewater collection system infrastructure research needs in the USA. *Urban Water* **4** (1), 21–29.

Teng, S., Chen, G. F., Gong, P. P., Liu, G. & Cui, F. S. 2020 Structural damage detection using convolutional neural networks combining strain energy and dynamic response. *Meccanica* **55** (4), 945–959.

Teng, S., Liu, Z. C., Chen, G. F. & Cheng, L. 2021 Concrete crack detection based on well-known feature extractor model and the YOLO_v2 network. *Applied Sciences-Basel* **11** (2), 13.

Wirahadikusumah, R., Abraham, D. & Iseley, T. 2001 Challenging issues in modeling deterioration of combined sewers. *Journal of Infrastructure Systems* **7** (2), 77–84.

Xiao, Q., Wang, J., Chen, H., Ye, S. & Xiang, L. 2019 The detection and evaluation by CCTU and rehabilitation analysis of sewer pipeline in an area of Shenzhen City. *Water & Wastewater Engineering* **45** (9), 109–114.

Xie, J. Q., Chen, H., Liao, Z. L., Gu, X. Y., Zhu, D. J. & Zhang, J. 2017 An integrated assessment of urban flooding mitigation strategies for robust decision making. *Environmental Modelling & Software* **95**, 143–155.

Xie, Q., Li, D., Xu, J., Yu, Z. & Wang, J. 2019 Automatic detection and classification of sewer defects via hierarchical deep learning. *IEEE Transactions on Automation Science and Engineering* **16** (4), 1836–1847.

Xu, J., Gui, C. Q. & Han, Q. H. 2020 Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering* **35** (10), 1160–1174.

Yin, X., Chen, Y., Bouferguene, A., Zaman, H., Al-Hussein, M. & Kurach, L. 2020 A deep learning-based framework for an automated defect detection system for sewer pipes. *Automation in Construction* **109**, 102967.

Zhang, X., Wang, Z., Liu, D. & Ling, Q. 2019 DADA: deep adversarial data augmentation for extremely low data regime classification. In: *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 2807–2811.

Zhang, C. B., Chang, C. C. & Jamshidi, M. 2020 Concrete bridge surface damage detection using a single-stage detector. *Computer-Aided Civil and Infrastructure Engineering* **35** (4), 389–409.

Zhou, Q., Leng, G., Su, J. & Ren, Y. 2019 Comparison of urbanization and climate change impacts on urban flood volumes: importance of urban planning and drainage adaptation. *Science of the Total Environment* **658**, 24–33.

Zhou, Q., Situ, Z., Teng, S. & Chen, G. 2021 Convolutional neural networks-based model for automated sewer defects detection and classification. *Journal of Water Resources Planning and Management* **147** (7), 04021036.

First received 15 October 2021; accepted in revised form 14 February 2022. Available online 1 March 2022