Automation for sewer pipe assessment: CCTV video interpretation algorithm and sewer pipe video assessment (SPVA) system development

Xianfei Yin\textsuperscript{a},\textsuperscript{*}, Tianxin Ma\textsuperscript{b}, Ahmed Bouferguene\textsuperscript{c}, Mohamed Al-Hussein\textsuperscript{d}

\textsuperscript{a} Department of Construction Management and Engineering, University of Twente, Enschede, the Netherlands
\textsuperscript{b} Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada
\textsuperscript{c} Campus Saint-Jean, University of Alberta, Edmonton, Alberta, Canada
\textsuperscript{d} Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta, Canada

\textbf{ARTICLE INFO}

Keywords:
Sewer pipe
CCTV
Assessment
Maintenance
Video interpretation
Automation

\textbf{ABSTRACT}

This research aims at improving the automation of the sewer pipe assessment process, specifically in terms of the development of a closed-circuit television (CCTV) video interpretation algorithm and sewer pipe video assessment (SPVA) system. A novel video interpretation algorithm for sewer pipes (VIASP) is proposed to use the labeled video (which is labeled by an automated defect detector) as the input in order to extract the useful information from the video, with the final output being the sewer pipe assessment report in textual format. To develop the VIASP, an optimization algorithm using simulated annealing (SA) is employed to determine the optimal human-defined parameters for the VIASP. A prototype of the SPVA system is developed to show how the developed automation techniques can fit into the daily workflow of sewer pipe assessment work. The effectiveness of the proposed method is validated in a case study.

1. Introduction

As the health condition of sewer pipes deteriorates as the pipes age, preventive maintenance must be carried out on a regular basis to keep the sewer pipes functional at all times \cite{5,41,44}. In current practice, closed-circuit television (CCTV) inspection is one of the most commonly used preventive maintenance techniques \cite{1}, which aims to detect all types of defects as early as possible to maintain sewer pipes in good health conditions. CCTV inspection techniques have been used for more than 40 years \cite{12}, and their wide application in the context of sewer pipe inspection is due to several reasons, such as CCTV inspection being safer to operate compared with the conventional man entry method \cite{29}, and the visual output being easier to understand compared with other techniques (e.g., laser-based system, ultrasonic-based sensors, etc.) \cite{4}.

The CCTV inspection process investigated in this research contains two main processes: on-site video collection and off-site video assessment (see Fig. 1). The on-site crew is responsible for collecting CCTV videos of sewer pipes based on the work order issued by the maintenance planning office, as indicated in Parts (a) and (b) of Fig. 1. Collected videos are then sent to an off-site office where technologists (trained individuals) need to watch through every one of them and denote every defect (and construction feature) that appears in the video according to a certain nomenclature, for example PACP \cite{28}, and WRc \cite{40}. As shown in Part (d) of Fig. 1, examples of the defect types (and construction feature) found in CCTV inspection video footage and their labels can include broken, hole, deposits, crack, fracture, root, and tap (construction feature). Based on the assessment results from the technologists, the sewer pipe inspection database can be constructed, where inspection information is stored for future maintenance planning purposes. The inspection information is presented graphically in Part (e) of Fig. 1, which includes the type of the defect (construction feature) and its location (distance to the start manhole in meters). In practice, the information pertaining to the defect is stored in a tabulated format in an Excel file or other database format (e.g., Microsoft Access). Table 1 shows a tabulated example of Part (e) in Fig. 1. In reviewing the current literature, defect detection, as shown in Parts (c) and (d) in Fig. 1, can be accomplished automatically by various methods, of which deep learning techniques have been explored extensively (e.g., \cite{1,22,45}). However, to get the final output of the CCTV inspection as shown in Table 1, further operations are needed. For example, technologists must still record the detected defect type and its location into an Excel file. This research aims at taking the automation of the CCTV inspection process a step further by interpreting the CCTV video and compiling the output

\textsuperscript{*} Corresponding author.
E-mail addresses: x.yin@utwente.nl (X. Yin), tianxin3@ualberta.ca (T. Ma), ahmedb@ualberta.ca (A. Bouferguene), malhussein@ualberta.ca (M. Al-Hussein).
information in a tabulated format as shown in Table 1, which is to automate the process from Part (b) to Part (e) in Fig. 1. To accomplish this, several challenges need to be overcome, and these are listed below.

1. In order for a defect detection algorithm to be considered effective, it must be outstanding in terms of both accuracy and processing speed. A highly accurate algorithm is the foundation for accurate video interpretation and a fast processing speed is what enables the algorithm to process video in real-time. The defects of the pipes will be labeled automatically by this defect detection algorithm, which will generate labeled videos as an output.

2. An interpretation algorithm must translate the information from labeled videos to tabulated text information. Note that, to date, none of the research available in the literature claims that a deep learning-based object detection algorithm could achieve a detection accuracy of 100%, because the deep learning-based algorithms have their own limitations and cannot achieve 100% accuracy. Therefore, deep learning-based defect detection for sewer pipes cannot attain 100% accuracy in practice. In fact, for certain images (i.e., video frames), the signal to noise ratio can be very low, which makes accurate detection and classification of sewer defects very challenging and at times erroneous. The inevitable errors generated by the object detection algorithm means it would be undesirable to translate the information directly from the labeled video to a textual output. The imperfect deep learning algorithm developed previously cannot achieve 100% detection accuracy for every image (i.e., video frame), which generates a certain amount of noise that could have a significant impact on the final textual output. This issue will be further discussed in Sections 3 & 4.1.

3. A text recognition algorithm must be able to extract the textual information contained in the video frame to identify the location of each detected defect or feature.

Many research efforts have been devoted to solving Challenge 1, which is to develop defect detection algorithms that are capable of detecting the defects within the video frames. State-of-the-art defect detection algorithms are reviewed in Section 2. In addition, our research group also developed a defect detection framework based on the algorithm called You Only Look Once (YOLO) [45]. The present study is built on the framework that we developed previously. There are also numerous studies targeting Challenge 3 (e.g., [11]), which are trying to recognize the text information within the CCTV video frames. This present study takes advantage of the existing text recognition technology developed by Microsoft to test our proposed software. Challenge 2 has not previously been investigated in a thorough manner and is the main target of this study. The proposed algorithm to solve Challenge 2 is described and validated in Section 4. Existing literature pertaining to Challenges 1 and 2 are reviewed and summarized in Section 2.

2. Background

Recognized for its ability to process high-dimensional natural raw data (e.g., image, speech), deep learning has made substantial breakthroughs in many areas (e.g., image recognition, speech recognition, natural language processing) for the development of artificial intelligence [21]. With the development of deep learning technology, computer vision techniques have been adopted in various areas, such as object detection, face recognition, action and activity recognition, human pose estimation, etc. [38], to improve the working productivity or level of automation. Object detection is used, in the context of defect detection for sewer pipe videos, to label the specific objects (e.g., crack, fracture, holes, etc.) that are typically identified by technologists, and further to automate this tedious and time-consuming process. Various frameworks have been adopted to facilitate object detection, among them, convolutional neural network (CNN) is the most widely used [19]. In recent years, several frameworks built on the original CNN structure have been proposed to improve the performance of object detection, such as region-based CNN (R-CNN) [7], Fast R-CNN [6], SPP-net [14], Single Shot Detector (SSD) [23], YOLO [31], Faster R-CNN [33], Mask R-CNN [13], etc. These various algorithms sought to improve detection accuracy and detection speed. Interestingly, these advanced object detection algorithms that have been investigated and developed by computer scientists are well adapted to solving practical problems occurring in construction engineering and management, such as construction safety and personnel monitoring, resource tracking, activity monitoring, surveying, as-is modeling, and inspection and condition assessment.

![Fig. 1. CCTV inspection workflow for preventive maintenance of sewer pipes.](image-url)
For the specific area of defect detection that is CCTV videos for sewer pipes, several studies have been carried out in the past few years. For example, Cheng & Wang [1] have adopted Faster R-CNN for the defect detection of sewer pipes and have achieved an mean average precision (mAP) of 83% for detecting roots, cracks, infiltration, and deposits; Kumar et al. [20] have employed a deep CNN to detect root intrusion, deposits, and cracks by using a dataset that contains 12,000 images, and reaching an accuracy of 86.2%; and Li et al. [22] have used deep CNN and hierarchical classification approach to classify deposit settlement, joint offset, broken, obstacles, water level sag and deformation within the sewer pipes. Kumar et al. [20] have developed an automated defect detection tool to detect root intrusion and deposits. In their research, the performance of defect detection frameworks Faster R-CNN, YOLOv3, and SSD are compared, showing that YOLOv3 has a more balanced performance in terms of detection speed and accuracy. Moradi et al. [27] have proposed a method to automate anomaly detection and localization in sewer CCTV inspection videos using multiple techniques such as support vector machine, maximally stable extremal regions algorithm, and CNN. Wang & Cheng [39] have developed a method using dilated CNN to retrieve segmentation of crack, deposit, and tree root defects from sewer pipe CCTV video frames. Dang et al. [3] have proposed a framework for facilitating the assessment of CCTV videos using the text recognition method. Hassan et al. [11] have used a CNN together with text recognition to generate defect reports based on keyframes (i.e., the frames that contain targeted defects) of the CCTV video, and Yin et al. [45] have proposed a deep learning-based framework to realize realtime defect detection for CCTV videos of sewer pipes, where YOLOv3 [32] is used as the defect detector. Information/data flows for such a framework have been streamlined, which could be a benchmark for future defect detection system development. Among them, all the studies focus on processing specific frames of the video, which is different from the video interpretation process. Note that video interpretation is different from processing every frame of the video, since there are further operations (e.g., noises need to be excluded since the deep learning algorithms cannot achieve 100% accuracy) need to be conducted before outputting the video information. The challenges and mechanism of CCTV video interpretation will be described in Section 4.1.

Video is a type of media that is much more information-intensive compared to images; however, fewer research efforts have been devoted to the processing (e.g., classification) of videos directly [17]. Although a video is essentially a stack of images, the information delivered in video format is different from static images since there is more contextual information. In the area of construction, research efforts have been made in terms of retrieving information from on-site surveillance videos. For example, Son et al. [37] have proposed a real-time warning system to prevent potential collisions between workers and heavy equipment based on real-time surveillance videos, where Faster R-CNN is used to detect the workers and 3D estimation is used to estimate the distance from workers to the equipment; Roberts & Golparvar-Fard [34] proposed an innovative method to track and analyze earthmoving activities based on videos using multiple machine learning algorithms, such as CNN, hidden Markov model, Gaussian mixture model, and support vector machine; and Gong & Caldas [8] proposed an intelligent video interpretation system for the productivity analysis of construction operations. The above-mentioned studies show that various types of valuable information can be retrieved from videos; however, in the sewer pipe maintenance process, few studies have focused on collecting information from CCTV videos directly, which is essential in the context of automating the process from Part (b) to Part (e) as shown in Fig. 1. Thus, the present research focuses its efforts on this research gap, which is to take the automation a step further than image processing-based defect detection to interpret the CCTV video and translate the information from video format to text format. The next section will describe the proposed method in detail.

1. No defects appear in these two frames. However, it is nearly impossible for such a small amount of time (i.e., approximately 0.08 s) pass between two individual defects since the camera is traveling at a speed of approximately 9 m/min (15 cm/s), which is the recommend speed from PACP code. Note that the actual traveling speed of the camera truck in our case study is even faster than 9 m/min, achieving at 23.6 m/min, according to one of the research that conducted by our research group [42].

2. The perspective from which the defect is viewed by the camera changes as the camera moves forward. In this case, these two frames with different camera angles may cause the defect detector to miss the defect, which could occur because these defect or features are unfamiliar to the defect detector at those particular camera angles.

3. Mistakes made by the detector. As mentioned above, automated defect detection cannot ensure 100% accuracy, and there is always the possibility that the detector will make some mistakes of either missing some defects, wrongly classifying a defect (e.g., classify a
crack as a root), or raising a false alarm (i.e., reporting a defect on the frame when there is not a defect present).

In addition to the factors mentioned above, other factors could also cause the discontinuity problem. For instance, the quality of the video changes during a short period of time. The quality of the video can be influenced by many factors such as lighting conditions, water that submerges the camera, mist or fog due to the temperature difference between the pipe interior and the ground, etc. Therefore, it is difficult to ensure the video footage is captured under the exact same conditions from beginning to end for each sewer pipe.

Given that so many factors can cause noise that hinders the performance of automated defect detection from CCTV videos, it is practically impossible (with currently available technology) to develop a perfect defect detector that makes no mistakes when analyzing video frames. To output the detection results frame by frame will generate noises, duplications, or false alarms in this case. Therefore, a VIASP is proposed to process the original output from the defect detector in a log file format (see Fig. 3) into output in tabulated format (textual) that contains information that is useful for the sewer pipe assessment department. The VIASP is described in more detail in Section 4.

After the log file generated by the defect detector is processed by the VIASP, a text recognition function, or optical character reader (OCR), is needed to extract the text information from the video frame that indicates the location to determine the distance of the detected defect from the starting manhole, as illustrated in Part (e) of Fig. 1. This function has been investigated by Dang et al. [3] by using Tesseract OCR to extract the text information (which included words in Korean that record the defect type and numbers that record the defect distance) from the CCTV video for sewer pipes. In their research, both the distance information and the defect type are included in the text information of the CCTV video, which differs from the video being processed in the present study since only distance information is included in the video footage under study and the defect type is determined by technologists working at the off-site assessment office (see the maintenance workflow presented in Fig. 1). Since there is no defect type information presented in text format in the video frame, instead the defect is automatically detected and labeled by the defect detector. In the present study, we use an OCR developed by Microsoft Azure (which is a cloud computing platform) to detect the text information within the video frame. The adopted OCR has a detecting accuracy of 90% on our dataset based on a prior experiment using 184 randomly sampled video frames. In this prior experiment, the errors largely occurred in one specific video where the color of the text was very close to the color of the background; therefore, the text in these frames may have been missed by the OCR. This problem can be improved by detecting the missed frames multiple times or replacing them with their adjacent video frames. With this implementation mentioned above, the detecting accuracy could increase to approximately 95%. A recent study by Moradi et al. [27] developed a text recognition system targeting sewer videos that performs well in solving the text background occlusion problem. The interested reader may consult the study by Moradi et al. [27] for more detailed information.

Note that the text recognition function is functionality that could be integrated into the proposed sewer pipe video assessment (SPVA) system. Future developers are free to use any functional OCR (e.g., Tesseract, Microsoft Azure, etc.) in the development of their software. In the present study, we chose to use the Computer Vision service (Microsoft, 2020) from Microsoft Azure since this OCR has been well investigated by previous researchers and useful tools have been developed and adopted to extract the text information contained within CCTV videos for sewer pipes [3]. The present research will focus on the development of VIASP and the overall SPVA system. With all the functions mentioned above, as shown in Parts (a)–(f) of Fig. 2, the information extracted from each step will be integrated and formatted as output in an Excel file, as per Part (g) of Fig. 2.

The final step is to package all the functions from Parts (a)–(g) of Fig. 2 into a user-friendly software, which is the proposed SPVA system. In addition to integrating all the functions including defect detection, video interpretation, and text recognition, the development of SPVA also takes into consideration the maintenance department’s management process to develop several useful functions in our case study. Based on the SPVA system developed for our case study, other developers could...
easily customize the system to fit their particular needs. The SPVA system is described in detail in Section 5.

4. CCTV video interpretation

4.1. CCTV video interpretation algorithm

A non-perfect deep learning algorithm could cause problems when trying to analyze the raw output of the defect detector. As in the example mentioned in the methodology section, any mistakes made by the algorithm could cause noise in the output. Theoretically, frames 1395–1403 could be considered as one defect as mentioned above (see Fig. 3); however, the two discontinuous frames make the situation more complicated. Instead of outputting one defect from frames 1395–1403 as one deposit, the raw output will be “deposit from 1395–1400”, and “another deposit on frame 1403”. In fact, one defect typically does appear in the video for approximately 1–4 s as the camera is moving forward, which will result in approximately 25–120 frames that contain the same defect. There is a possibility that the non-perfect defect detector makes several mistakes for these continuous frames due to the reasons previously discussed in the methodology section. If a random number of mistakes occur at random frames in a continuous series of frames, the one defect will be cut into little pieces, instead of showing as one defect that appears in continuous frames. For example, Fig. 4 presents an example of the consequence of this noise that can then be found in the raw output of the defect detection log file. There are cracks that are detected in frames 100–108, 110, 113–140, one fracture on frame 109, one deposit on 112, and no defects on frame 111. For a video with a frame rate of 25 FPS, each of those frames in Fig. 4 will appear in the video for 0.04 s. In fact, considering the short time period during which these video frames appear on the screen, it is likely that from frame 100 to frame 140, there is only one defect, i.e., crack. Fig. 4 illustrates an extreme example that would occur very rarely considering the accuracy of the defect detector that we employed in the present study. Note that even if the defect detector has an excellent detection accuracy, the unpredictable underground conditions during video collection process will decrease the performance of the defect detector to some extent. Possible factors that influence the performance of the defect detector are mentioned in the methodology section, such as a change in video quality or change in camera angle, etc.

Several common scenarios found in the labeled CCTV videos are shown in Fig. 5. Scenario 1 is the ideal situation where continuous frames are detected with only one type of defect, which can easily result
in output indicating there is a defect, which is crack in the case of Scenario 1 in Fig. 5, from frame $i$ to $i+n$. Scenario 2 is more complicated than Scenario 1 since there is noise in the series of frames, which could result in output indicating multiple different defects as described in Fig. 4. Scenario 3 is the most complicated situation in which a small number of no defect frames (or frames that contain defects with a low confidence score) together with some noise are included in the series of continuous frames from frame $i$ to $i+n$. The gap (frames with no defects) could split the continuous frame cluster into smaller pieces, which will increase the level of difficulty for interpreting the video. In an ideal situation, all the scenarios presented in Fig. 5 should result in the same output, which is that there is a crack from frame $i$ to $i+n$. In order to accomplish this, VIASP is proposed to exclude the noise and merge frames that show the same defect.

Before describing the VIASP in detail, the concept of the defect frame cluster (DFC) is proposed:

- For a certain number of continuous frames, defects (that could be of any type) are detected in almost all these frames. If the frame gap ($g$) between those adjacent frames that do contain defects is less than a predefined parameter $G$ (see Fig. 5), then consider the continuous frames that come before and after the gap together as one cluster.

Based on the proposed concept, the three scenarios presented in Fig. 5 contain one defect frame cluster (DFC) each. The output of the VIASP is the information about this DFC, such as defect type, the average confidence score of the DFC, and start frame and end frame of the DFC.

The defect type for the DFC is the defect type that appears most frequently in one DFC (majority rule). For example, for a DFC with 100 frames, of which 95 are detected with a crack and 5 are noise frames (either with other types of defects or are frames with no defects), then the defect type for this DFC is determined as crack. The average confidence score of this DFC is calculated based on those frames that show the same type of defect as the DFC.

The pseudo-code for the VIASP is presented in Fig. 6. The algorithm starts by defining the initialized variables and the human predefined variables. The initialized variables include the counting variable for frames ($i$) and the counting variable for the DFC ($j$). The predefined parameters are merge gap ($G$) (see $G$ in Fig. 5), which is used for determining whether discontinuous frames are within the former DFC; confidence score $C_1$, which is used to filter out all the frames that are detected with defects and have confidence scores lower than $C_1$; confidence $C_2$, which is used to filter out all the DFCs that have confidence scores lower than $C_2$. The determination of these predefined parameters is discussed in Section 4.2. The log file generated by the defect detector needs to be loaded to VIASP to extract the information about the frames’ indexes, detected defect types, and confidence scores for the defects (Step 2 of Fig. 6).

After Step 2, the first-round confidence score filtering (see Step 3 of Fig. 6) aims to filter out all the frames that have a lower confidence score. For the frames with no defects and the frames with low-confidence-level defects, the VIASP assigns an index $T_i = 0$ to them, which means there are no defects detected on the $i^{th}$ frame. The results of the frames will be assigned with an index of $T_i$ according to different
types of defects. For example, if there is a crack detected on \(i^{th}\) frame, then a value of 4 is assigned, i.e., \(T_i = 4\).

After assigning an index of \(T_i\) to each frame of the video, the next step is to find the DFC of the video (see Step 4). To find the DFC, an important parameter, merge gap (G), is needed to determine whether the next frame with a defect after a gap (a gap of continuous frames with no defects) is within the DFC or not (for a graphical representation, see Fig. 5). After a continuous series of frames with no defects, the number of frames within this gap are counted, and if the count number is lower than G, then the next frame is within the DFC (\(j^{th}\) defect), otherwise the frame with defects is in the next DFC (\((j+1)^{th}\)).

After finding all the DFCs in the video, the type of each DFC needs to be determined based on the majority rule mentioned above. One exception is that if the DFC includes any taps (any frames for which \(T_i = 7\)) among the DFC, then define the type of \(j^{th}\) DFC as “Tap”.

Step 1. Start:
   a) Initialized variables: \(i=1\) (1\(^{st}\) frame), \(j=1\) (1\(^{st}\) defects);
   b) Predefined variables: merge gap (G), confidence 1 (C1) and confidence 2 (C2);
Step 2. Read the log file generated by YOLO;
Step 3. Assign an index (\(T_i\)) for all the frames:
   a) If there is no defect: \(T_i = 0\);
   b) If there is a defect, then:
      i) If the confidence score is smaller than C1 (\(c < C_1\)), then \(T_i = 0\);
      ii) If the confidence score is bigger than C1 (\(c > C_1\)), then: Broken: \(T_i = 1\); Hole: \(T_i = 2\);
         Deposits: \(T_i = 3\); Crack: \(T_i = 4\); Fracture: \(T_i = 5\); Root: \(T_i = 6\); Tap: \(T_i = 7\);
Step 4. While \(i <= \) total frames:
   a) Find the 1\(^{st}\) \(T_i > 0\), and count it as the 1\(^{st}\) defect, \(j=1\);
   b) Find the DFC (within the predefined merge gap G) that include this frame;
   c) Output defect type for \(j^{th}\) DFC
      i) If there is \(T_i = 7\) among the DFC, then define the type of \(j^{th}\) DFC as “Tap”
      ii) If there isn’t \(T_i = 7\) among the DFC, then define the type of \(j^{th}\) DFC as the most frequent defect type (DTj) among the cluster;
   d) Calculate the average confidence score for all the defect= DTj in this cluster as the confidence score (CSj) of \(j^{th}\) DFC;
   e) If the confidence score (CSj) of \(j^{th}\) DFC is smaller than C2, then discard it; Otherwise, keep it;
   f) Record defect type, confidence score, start frame, and end frame for \(j^{th}\) defect;
   g) Find the next \(T_i > 0\), and \(j=j+1\);
Step 5. Format the output in the EXCEL file;
Step 6. End

Fig. 6. Pseudo-code for the VIASP.

Fig. 5. After a continuous series of frames with no defects, the number of frames within this gap are counted, and if the count number is lower than G, then the next frame is within the DFC (\(j^{th}\)), otherwise the frame with defects is in the next DFC (\((j+1)^{th}\)).

Fig. 7. CCTV recording for tap within the sewer pipe.
are images of defects within tap A, the defect detector will detect them, recorded in the manual sewer pipe assessment report; however, if there are usually not recorded by the technologist in the sewer pipe assessment or VIASP) due to various reasons. For example, there exists an inconsistency in the manual assessment method since different technologists may have slightly different criteria according to which they categorize those defects that are closely related (e.g., crack and fracture). As for the VIASP, the imperfect defect detector is the reason the VIASP does not have perfect accuracy. Therefore, the objective is to select values for \( C_1, C_2, \) and \( G \) that make the output of the VIASP as close to the manual assessment report as possible since the manual assessment is the only existing dataset against which the results can be compared and validated. To realize the objective, a searching method needs to be carried out to find the optimal combination of these three predefined parameters. To match the output of the VIASP with the manual assessment report directly is difficult at this stage, and the reason for that is described in the following section. If matching the defects outputted by the VIASP and the assessment report is impossible (or cost-prohibitive), we can match the number of DFCs outputted by the VIASP and the number of defects in the manual assessment report to facilitate finding the optimal predefined parameters, \( C_1, C_2, \) and \( G \). Therefore, our objective function for the searching algorithm is to minimize the difference between the number of the DFCs outputted by the VIASP and the quantity of defects in the manual assessment report. The searching algorithm for finding the predefined parameters is described in the following section (see Section 4.2.2). The performance of the VIASP with the selected predefined parameters is also tested in our case study to ensure the algorithm is accurate (see Section 4.2.3).

The comparison of the output of the VIASP with the manual assessment report cannot be conducted automatically at this stage since the location of the defect identified by the VIASP is different from the location indicated in the manual assessment report. For example, if there is a defect at 2 m from the starting manhole, the technologist conducting the manual assessment will wait until the defect has just disappeared from the screen before recording the location since this is the location at which the defect is right beside the crawler. However, the VIASP will output the starting frame and the end frame of the DFC representing the same defect. The location (in meters) shown on the starting frame is different from the real location of the defect since the meters indicated on that frame is ahead of the real location of the defect. As for the end frame of the DFC, this also varies from the real location in certain circumstances. As the camera is moving, the defect will disappear from the screen gradually, and during the process, the perspectives at which the defect is viewed keep changing, which may lead to the end frame of the DFC not aligning with the very last frame before the defect disappears from the video completely. In addition, sometimes two adjacent defects are very close to each other, which makes it difficult to discriminate one from the other. In summary, it is difficult to match a defect found by way of the VIASP with the defect found via manual assessment automatically. Therefore, the present study proposes to compare the number of defects identified by VIASP with the number of defects identified by technologists conducting a manual assessment instead of matching the output of the VIASP with the manual assessment report directly. The objective function is presented in Eq. (1).

\[
\text{Obj} = \min\{C_1, C_2, G\} = \text{Num}_{\text{VIASP}}(C_1, C_2, G) - \text{Num}_{\text{manual}}(\text{defects})
\]

where \( R(C_1, C_2, G) \) is a function used to calculate the difference between the number of defects identified by VIASP and the number of defects contained in the manual assessment report. \( \text{Num}_{\text{VIASP}}(C_1, C_2, G) \) is the number of defects identified by VIASP with the predefined parameters of \( C_1, C_2, \) and \( G \). \( \text{Num}_{\text{manual}}(\text{defects}) \) is the number of defects in the manual assessment report for our testing dataset, which is a constant number once we select a dataset.

The objective function is subject to Eqs. (2)–(6):

\[
\text{Num}_{\text{VIASP}}(C_1, C_2, G) - \text{Num}_{\text{manual}}(\text{defects}) \geq 0
\]

\[
0 \leq C_1 \leq 100
\]

\[
0 \leq C_2 \leq 100
\]

\[
0 \leq G \leq 300
\]
Eq. (2) shows that the Num_{DPC}(C_1, C_2, G) is greater than Num_{DPC}\_manual, which means that the number of DPCs identified from VIASP should not be less than the number of defects in the manual assessment report. The reason to set this constraint is that the risk of false alarm is lower than the risk of missing a defect, thus, the VIASP tries to cover as much of the defects as possible, while outputting fewer false alarms. The constraints for C_1 and C_2 range from a no-confidence threshold (confidence score equal to 0) to the highest confidence threshold (confidence score equal to 100). The merge gap (G) is set to 0–300, which is 0–12 s of video time considering the frame rate property of the video (25–30FPS), which means that the DPC could have a duration in the range of 0–12 s. The constraint for G is manually set to be a larger range to include more possibilities for the next step (optimization algorithm). In addition, a large range for G facilitates the inclusion of a larger DFC, such as the DFC for a tap. Eq. (6) shows that C_1 must be less than or equal to C_2, otherwise C_2 is meaningless, since if C_1 is higher than C_2, then the confidence for every DFC must be higher than C_2, which means C_2 cannot filter out any DFCs. With the objective function and constraints listed from Eqs. (1)–(6), an optimization algorithm is developed as described in the following section to find the combination of C_1, C_2, and G that leads to an optimal solution of minf(C_1, C_2, G).

4.2.2. Optimization algorithm

Optimization, which is central to decision making in various areas (such as engineering and economics), aims to find the best solution from among many possible alternatives [2]. To find the best solution of C_1, C_2, G from all the possible alternatives, an optimization algorithm called simulated annealing (SA) is employed in this study. The SA algorithm, first proposed by Kirkpatrick et al. [18] in the category of random searching algorithms [46], is developed to solve complicated combinatorial optimization problems [16], and is known for preventing the algorithm from getting stuck in a local minimum point in the searching process [2]. As we can see from Fig. 8, the global minimal point is A, while B is the local minimizer. If the random starting point is C, the naive random searching algorithm [2] will move toward point B and stick with it as the minimal point, since it is hard for naive random searching algorithm to climb over the local maximum point D. While the SA algorithm is designed to climb out of the local minimum point (point B in Fig. 8) and to try to find the global minimum point (point A of Fig. 8) as the result. The interested reader may consult Chong & Zak [2] for a detailed description of the mechanism underlying the naive random searching algorithm and SA. The SA algorithm has been successfully employed in many optimization problems, for example, Paya et al. [30] applied SA to solve multi-objective optimization problems of concrete frame design; Zeferino et al. [47] used SA for wastewater system planning; and Hackl et al. [9] employed SA in planning restoration programs for transportation networks. A detailed description of the SA algorithm is provided in the following section.

The pseudo-code for the SA to find the minimal point for f(x) is presented in Fig. 9. Step 1 is to randomly sample a point from Ω, which is the domain of the function, as the starting point from where the algorithm starts its searching process, which is to sample a point from its neighborhood NC(x^{(k)}). Step 3 defines the rule for whether to accept a candidate point or not. If a candidate point is a better point (the value of the objective function is smaller), then accept the point as the next point. If the candidate point is a worse point (the value of the objective function is larger), then the algorithm will accept the worse point with a probability of p(k,f(x^{(k)}),f(x^{(k)})), which is calculated by \(\exp\left(\frac{-\Delta E}{k}\right)\).

where \(\Delta E\) is the energy difference between the candidate point and the original point, \(T_k\) is the temperature of the \(k^{th}\) iteration. As the number of iterations increases, \(T_k\) decreases, which ensures that the probability of accepting a worse candidate point will decrease as the algorithm runs to a higher number of iterations, which is a process that mimics the physical annealing process for hot metal at high temperature changing to low temperature [9,26]. This rule can help the algorithm avoid getting stuck in one local minimum point, and always provides the algorithm with the possibility to move to the global minimum point. After each iteration, the best-so-far point is recorded in Step 4. Stop criteria is checked in the next steps to determine whether to stop the algorithm or return to Step 2 to start another iteration.

Parameter selection is determined according to several factors, such as previous studies, the property of the problem, and experiment performance. Since the SA algorithm is a heuristic algorithm, it does not have only one true solution. The design of such an algorithm is problem-specific. In our case study, parameters (or the structure) of the SA algorithm, such as neighborhood selection, cooling schedule, initial point, and stopping criteria are determined based on the factors mentioned above. Validation of the results of the optimization algorithm is described in Section 4.2.3. Neighborhood selection is mainly a problem-specific choice [9], which requires, to some extent, subjective judgment that considers the particular situation related to the question itself. In the present problem, which considers the feasible range of three predefined variables (as mentioned in Section 4.2.1), the neighborhood is sampled from integer numbers from \([-5,5]\), since all three variables are integer numbers in our case. There are many previous studies focused on the cooling schedule, such as Hajek [10] and Siddique & Adeli [36]. The method used in the present study was proposed by Johnson et al. [15] to define the initial temperature \(T_0\) and the minimum temperature at the last iteration \(n\) \(T_n\). The method has been adapted in many studies, such as Zeferino et al. [47]. With the initial acceptance rate \(p_0\) and the last acceptance rate \(p_n\), \(\hat{T}_0\) and \(\hat{T}_n\) can be calculated as \(\hat{T}_0 = \frac{1}{\ln(p_0)}\) and \(\hat{T}_n = \frac{1}{\ln(p_n)}\). We choose a typical initial acceptance rate of 0.8 and final acceptance rate of 0.001, which means the probability for accepting a worse candidate point is higher at the early stage and much lower at the final stage. \(T_0\) and \(T_n\) can be calculated accordingly. The decay factor that controls how the temperature changes at each step can be calculated as \(\frac{T_{k+1}}{T_k} = \frac{\hat{T}_{k+1}}{\hat{T}_k}\). The next iteration temperature can be calculated as the current temperature multiplied by this decay factor, which simulates the annealing process in the field of metallurgy. For the initial point, we use our best educated guess \([C_1, C_2, G]_0 = [20,30,30]\), which is informed based on the experience of watching CCTV videos and reviewing their raw log files generated by the defect detector. A fixed number of iterations (100) is used as the stopping criteria, and we can always increase the number of iterations based on the output of the algorithm, and the best-so-far point will be recorded during the whole process.

The implementation of the SA with the objective function and
The objective function keeps decreasing with several small fluctuations during the process and finally reaches a stable range around 0. The values of $C_1$, $C_2$, and $G$ during the searching process are plotted in Fig. 10 as well. The overall trend is contrary to the objective function, as the three variables keep increasing and finally stay stable at a certain range. The minimum value of the objective function is 4, which means the number of defects identified by the VIASP is 4 more than that of the manual assessment report. The best point is $[C_1, C_2, G]_{\text{best}} = [42, 80, 85]$, which means that the optimum solution generated by the SA is that $C_1$ is equal to 42, $C_2$ is equal to 80, and $G$ is equal to 85. These three values generated by the SA for the predefined parameters are then used in the VIASP to generate an assessment report automatically, and the performance of the VIASP using these predefined variables is tested in the next section.

4.2.3. Validation and analysis for VIASP

The objective of the optimization algorithm (i.e., SA) is to find the optimal solution for three predefined variables required by the VIASP, which are $[C_1, C_2, G]_{\text{best}} = [42, 80, 85]$. With these three variables and the log file generated by the defect detector, the video assessment report can be generated automatically by VIASP. Fig. 11 shows the results of the comparison of the output of VIASP and manual assessment results. The matched defects are shaded with a green background in Fig. 11 and linked with the two-way arrows. There is a small difference in terms of the distance (shown in meters) between the two results. The VIASP generates two distances for one defect: start distance and end distance. However, there is only one distance identified by technologists and recorded in the manual assessment report. The start distance is always ahead of the distance indicated by manual assessment result, which occurs because once the defect appears on the screen, the defect will be labeled. The end distance is relatively close to the distance indicated in the manual assessment report, although there are small differences. Three false alarms are reported by the VIASP, and two cracks are missed by the VIASP in this example.

The performance of the VIASP is tested with five CCTV videos including 98 defects identified in the manual assessment report in total. The performance of VIASP is tested by comparing the output of VIASP with the manual assessment report. The comparison is conducted manually since the matching of the defects from VIASP with the manual assessment report automatically is challenging at this stage (as described in Section 4.2.1). F-measure, which is a widely used metric to test accuracy of models/algorithms in classification problem [24, 35], is used in this study to measure the accuracy of VIASP. The definitions of F-measure are presented in Eqs. (7)–(9).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}

where TP is the true positive result, FP is the false positive, FN is the false negative. Based on the equations, precision denotes the correct rate over all the detected objects, while recall is the rate of completeness of detected objects among all the manual results [45]. A detailed description of F-measure can be found in a study authored by Sasaki [35]. Overall, for the selected five videos with 98 defects, the precision for VIASP is approximately 0.77 and the recall is approximately 0.73, which gives an F1 score of 0.75. There are areas for potential improvement in future research in terms of accuracy for the proposed algorithm. Several
examples of how real-life conditions could influence the performance of the VIASP are summarized below:

1. The inconsistency of the manual assessment results influences the value of the metric that measures the performance of the algorithm (i.e., $F_1$ score in this study). The manual denotation of some small defects, such as deposits, shows some inconsistency compared with the VIASP. For example, as is shown in Fig. 12a and Fig. 12b, for two similar deposits, one technologist recorded one as a deposit; however, a similar deposit was not recorded as a deposit by another technologist (or perhaps even by the same technologist). In addition, in some cases, small defects may be ignored or overlooked by the technologists (as shown in Fig. 12c, which is the No.1 defect in Fig. 11), but the same defects are detected by the defect detector because the VIASP has a consistent standard for identifying the defects. This issue has a negative impact on the $F_1$ score in our case, if the impact of the manual assessment method’s inherent inconsistencies could be accounted for, the measured performance of the algorithm would improve.

2. Some of the defects (e.g., deposits) within a tap are identified and outputted in the final report by the VIASP; however, these defects do not need to be recorded in the manual assessment report. For instance, Fig. 12d shows an example of deposits within a tap that were identified by the VIASP, but were not recorded in the manual assessment report. In practice, these defects are not counted as defects for the inspected sewer pipe because they are in tap. We developed a rule to avoid this situation (see Step 4-c of Fig. 6), which works for the majority of cases; however, a certain number of false alarms are still made by the VIASP. In fact, the false alarms for deposits (No. 8 and No. 17) in Fig. 11 are deposits within a tap.

3. Some minor defects are detected and labeled in the CCTV video; however, some of them are excluded from the final output of VIASP because these detected frames show a low confidence score, as shown for example in Fig. 12e. The crack is labeled by the defect detector successfully, but the confidence score of the specific frame or confidence score of DFC that represents this crack is lower than the confidence threshold that we defined. During the trade-off process of choosing the confidence score and the merge gap, a relatively higher confidence score is selected that results in the detection recall of some of the minor defects being sacrificed to eliminate a large number of duplications in the final report.

4. Obstructions within the sewer pipe that hinder the moving of the crawler will cause false alarms since the CCTV camera needs to rotate to different angles to check how the obstruction is blocking the path, which will generate many abnormal views for the defect detection algorithm. An example of the crawler being blocked by obstructions is presented in Fig. 12f. The operator is moving the camera to face the crawler’s wheel to check the condition of the obstruction. The defect detector wrongly detects the wheel as a hole in this case. It is difficult to detect anything precisely based on these views since these views are not included in the training dataset, which means that they have never before been seen by the defect detection algorithm. A potential direction for improving the performance of the VIASP would be to include more abnormal views of CCTV videos in the training of the defect detector in order for the defect detector to be able to discern these images, which will in turn improve the performance of the defect detector, and further to improve the performance of VIASP.

5. The quality of the video may change significantly due to human factors (improper operation such as moving too fast, not using lighting properly, etc.) or non-human-factors (water sag or smoking within the pipe) during the recording process. Fig. 12g shows the camera being submerged in water sag, which will cause discontinuous frames if these frames are within the middle of a DFC or cause false alarms. Fig. 12h shows the blurred video frame caused by unknown equipment problems.

In summary, taking into consideration how these factors influence the performance of the VIASP, the $F_1$ score of 0.75 for the proposed VIASP is considered reasonable, yet there is still potential for improvement. Future efforts could be made to improve the performance of the algorithm including enhancing the performance of the defect detector as mentioned above.

5. SPVA system prototype development

With the developed defect detector and VIASP, the present research develops a sewer pipe video assessment (SPVA) system prototype to pack all the developed functionality into a user-friendly software package to facilitate the daily operations of the sewer pipe maintenance department. Fig. 13 presents the interaction and workflow between the on-site and off-site parts of the overall process undertaken by a sewer pipe maintenance department. As is shown in the existing information flow in Fig. 13, the maintenance office can generate work orders for CCTV collection based on the maintenance schedule, which is developed based on the historical CCTV video assessment reports. CCTV collection crews will collect the CCTV videos, while in the meantime, log files recording daily works are generated along with the CCTV collection processes. With the log files, the assessment office can know which videos are most recently collected and still need to be assessed. Then, each of the newly collected videos is watched from the beginning to the end by a technologist. The assessment reports are generated while watching these videos. With the proposed SPVA system, the assessment process could be completed automatically (see the updated information flow in Fig. 13). The log file that records the daily work completed and the CCTV videos there were collected can be fed into the SPVA system directly. The SPVA system will find the targeted CCTV video and complete the assessment process automatically (defect detection, video interpretation, text recognition, writing to Excel, etc.). The generated report then can be sent to the assessment technologist for review and further analysis.

A prototype of the SPVA system, whose user interface (UI) is shown in Fig. 14, is proposed in this research. The UI is developed in Microsoft Visual Studio 2019. Part A of Fig. 14 is the menu bar, which includes File, View, Search, Train YOLO, and Setting (see Part A on the right of Fig. 14 for the submenus within each menu category). The user can open a specific CCTV video that needs to be processed from the File menu. After the processing of the video is complete, the user can choose to save the results in Excel format or Database format (e.g., Microsoft Access database), or both. The View function generates a summary of the process results. For example, information such as the total number of videos processed, or the total number of each type of defect can be viewed here. The Search function gives the user the option to search a specific type of defect within a specific location of the processed sewer pipe. The Train YOLO function facilitates the retraining process for the defect detector (i.e., YOLO in this case) by helping the user to select the image files and annotation files (e.g., .TXT file and .XML file) to replace the old versions of image files and annotation files. Setting function provides a portal to link the SPVA with Microsoft Azure for executing the text recognition function. Users could also set the saving path for the labeled video, and the generated Excel (or database) file.

Part B of Fig. 14 is the run YOLO module, where parameters such as confidence score (C1, C2), merge gap (G), weight (different versions of previous trained YOLO), could be set and adjusted by the user. The video number of the video currently being processed is showing after the label of video. Several parameters can be adjusted and the Confirm button loads the parameters that were changed by the users: for example, the weight button is for selecting a previously trained YOLO and loading its weight to the software, the Run button is to start the video processing, and the Analyze button can analyze the log file generated by YOLO using VIASP, instead of going through the process of running YOLO and then running the VIASP.

The multi-task module is shown in Part C of Fig. 14, where users can process a folder of videos instead of running a single video in module B.
The Video selector helps to filter the videos by date since the video database is typically contain a large number of files, thus, filtering could help users find the targeted videos more efficiently. The schedular module (Part D of Fig. 14) is a feature that can schedule the assessment of the videos automatically based on the uploaded CCTV inspection log files. The CCTV inspection crews will upload the log file once they are finished their daily work, and the log file is where such information as collecting time, media storage location, sewer pipe number, uploading time, etc., is recorded. The schedular module can identify those videos that have been uploaded but have not yet been assessed. Then, according to a set schedule (e.g., 6 am, from March 1 to March 30, daily), the schedular automatically starts the software and processes the newly uploaded videos according to the inspection log file. Assuming all the video processing jobs could be done within two hours, then, the assessment technologists could see the assessment report at 8 am when they arrive at the office. Therefore, this schedular feature could save time.
in terms of the amount of work the technologists must do when they arrive at the assessment office.

In Edmonton, CCTV videos are usually collected at night since there will be less disturbance for the traffic compared with the daytime collection. The collected videos are sent to the off-site assessment office in the early morning every day. The SPVA system can start processing newly collected video and finish the processing before the technologists arrive at the office. With the automated generated list of defects for all the newly collected videos, technologists are able to have a quick judgment of the health condition of the targeted pipes and schedule the pipes with severe defects at the top position in their priority list for manual assessment. In addition, with the defects list, technologists do not need to watch through every video to label defects manually, instead, they could fast forward videos to locations with defects indicated by the automatically generated reports. After locating the defects in videos, the PACP code still needs to be labeled by technologists manually. However, the assessment productivity will be improved with the facilitation of the SPVA, since the technologists do not need to watch through every collected video.

Note that this is a proposed prototype based on our case study, which aims specifically at facilitating the work processes in our case study. With the developed core functions (defect detection, video interpretation, etc.), the features in terms of application could always be adjusted for other cities’ municipal departments. This prototype is simply a showcase to demonstrate how to integrate the developed machine learning-based automation tools into real-life sewer pipe maintenance work. Admittedly, there is room for improvement in terms of software development; however, the backbones (e.g., defect detector, VIASP) of the SPVA system are generalized tools that could improve the automation of sewer pipe assessment work.

6. Discussion

A highly accurate defect detector is the foundation for the VIASP as well as the SPVA system, since the more accurate the detector is, the more accurate the VIASP will be. The accurate output of the VIASP enables the SPVA system to improve the level of automation of the overall sewer pipe assessment process. This study employed an excellent defect detector based on the YOLO algorithm; however, improvements could be achieved with more efficient and accurate algorithms in future research regarding defect detection. The performance of VIASP could be improved as a result of using a better defect detector; therefore, developing a more advanced defect detector is an important research direction that requires continuous efforts in the future.

Multiple prior experiments were conducted before finalizing the structure of the VIASP since the development process starts with a thorough understanding of the targeted problem and then considers the estimation of the performance of the algorithm for different versions of the VIASP during the algorithm design process. The algorithm development process is a trial and error process; lessons can be learned from all the previous efforts. In the future, the performance of VIASP could be improved in various ways, for example, by adding more filters to filter out the false alarms that cannot be filtered out at this stage (e.g., defects within the tap), or by finding the appropriate confidence score for each type of defect instead of using one confidence score for all the defects, as was done in this study.

This research offers four contributions to the body of knowledge. 1) This research proposes an automated sewer pipe video assessment framework that includes defect detector, video interpretation algorithm, text recognition function. This present study provides a framework that future similar system development can follow. 2) This study proposes a novel video interpretation algorithm for sewer pipes (VIASP), which is tested and validated in this study. An optimization algorithm is employed to find the optimal human-defined parameters for the VIASP. 3) The authors of the present study develop a prototype for the sewer pipe video assessment (SPVA) system, which demonstrates how the developed automated functions (defect detector, VIASP, etc.) can be deployed in a real-life sewer pipe maintenance scenario considering the workflow of the daily operations. Note that the VIASP can improve productivity, but at this point it is not a replacement for the technologists since there are mistakes made by the defect detector. In this case, risk management should be considered in the implementation of this kind of system. 4) Overall, the research improves the level of automation in sewer pipe assessment work by generating a textual summary of the assessment results for sewer pipe CCTV videos directly. Consistency is a by-product of the automation of the sewer pipe assessment system since the defect detector and the VIASP can output assessment results in-line with a constant standard.

The research has some limitations that need to be resolved in the future research. In this research, the manual assessment report generated by the sewer pipe experts serves as the ground truth; however, the manual assessment report is not 100% accurate due to human factors such as the variable performance, level of experience, level of fatigue, etc. Nevertheless, the manual assessment report is the best benchmark available in our experiment. Further improvement can be made by assembling an assessment team to generate the manual assessment report instead of using the assessment report generated by one individual pipe expert. In addition, promising directions to improve the performance of the proposed SPVA system including 1) retraining the defect detector to be able to identify non-forward-facing frames to eliminate the noises that affect the performance of the VIASP, 2) retraining the defect detector to be able to output PACP code directly for the detected defects, 3) incorporating frame rate reduction to improve the performance of the VIASP.

7. Conclusions

Sewer pipe assessment is the foundation on which an efficient maintenance plan is developed since the health condition of the sewer pipe is assessed and recorded as the basis for decision making in this process. In order to improve the level of automation of this CCTV-based sewer pipe assessment process, various previous studies have developed different versions of automated defect detectors for labeling the defects that appear in the video with specific names according to a particular nomenclature. However, the labeled videos or video frames require further processing to extract useful information from the CCTV videos and to automatically generate the assessment reports. The present research aims at improving the automation for sewer pipe assessment in terms of CCTV video interpretation and SPVA system development. The VIASP is designed to identify the frame clusters that could represent the defect in the sewer pipe. The VIASP features confidence score filtering, and the merging of frame gaps with no defect or discontinuous frames. Three key predefined parameters \( (G_1, G_2, G_{best} = [42, 80, 85]) \) for VIASP are determined by an optimization algorithm (i.e., SA). The performance of the VIASP is tested by comparing the output of the VIASP with the manual assessment results. The testing shows that the VIASP could realize an \( F_1 \) score of 0.75. Potential factors that could affect the performance of the VIASP are analyzed in detail, which provides insights for potential ways to improve the performance of VIASP in future works. In order to show how the previously developed automated functions (e.g., defect detection, video interpretation, text recognition) aimed at facilitating the sewer pipe assessment could fit into the everyday workflow of sewer pipe maintenance work, a prototype of an SPVA system is proposed to demonstrate the overall process. Backbone components (i.e., defect detector, VIASP, text recognition algorithm) of the SPVA are replaceable whenever more advanced techniques are available. Overall, this research could serve as a framework for future similar system development.

Declaration of Competing Interest

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

Acknowledgments

The authors would like to thank EPCOR Drainage Services based in Edmonton, Canada for their technical support and consultation, with special thanks to Randy Russell and Luke Kurach. We gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (CRDPJ 503647-16), and the National Natural Science Foundation of China (Grant No. 72002152).

References

[1] J.C.P. Cheng, M. Wang, Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques, Autom. Constr. 95 (2018) 155–171, https://doi.org/10.1016/j.autcon.2018.08.006.
[2] E.K.P. Chong, S.H. Zak, An Introduction to Optimization, Fourth edition, Wiley Series in Discrete Mathematics and Optimization, 76, Wiley, 2013 (https://doi.org/10.1002/9781118333460).
[3] T.M. Dang, S.I. Hassan, S. Im, M. Mehmood, H. Moon, Utilizing text recognition for the defects extraction in sewers CCTV inspection videos, Comput. Ind. (2018), https://doi.org/10.1016/j.compind.2018.03.020.
[4] D. Duran, K. Althoefer, L.D. Seneviratne, State of the art in sensor technologies for underground sewer inspections, Autom. Constr. 111 (2020) 103061, https://doi.org/10.1016/j.autcon.2019.102947.
[5] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. (1953), https://doi.org/10.1111/mice.12481.
[6] J. Redmon, S. Divvala, R. Girshick, A. Farli, You only look once: unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.
[7] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, in arXiv preprint arXiv:1804.02767, 2018, https://doi.org/10.1109/CVPR.2017.690.
[8] S. Kolender, D. Colloca, A. Boni, J. Kurash, Z. Seneviratne, State-of-the-art in sensor technologies for underground sewer inspections, Autom. Constr. 111 (2020) 103061, https://doi.org/10.1016/j.autcon.2019.102947.
[9] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Autom. Constr. 20 (8) (2011) 1211–1226, https://doi.org/10.1016/j.autcon.2011.05.005.
[10] S. Siddique, H. Adeli, Simulated annealing, its variants and engineering applications, Int. J. Artificial Intell. Tools (2016), https://doi.org/10.1142/S0218213016500156.
[11] J. Hacik, B.T. Adey, N. Lehan, Determination of near-optimal restoration programs for transportation networks following natural hazard events using simulated annealing, Comput. Aided. Civ. Infrastruct. Eng. 33 (8) (2018) 618–637, https://doi.org/10.1111/mice.12346.
[12] R. Navab-Kashani, F. Golkhou, Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition, J. Constr. Eng. Manag. 141 (6) (2015), 04015099, https://doi.org/10.1061/(ASCE)CE.1943-7862.0000976.
[13] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, J. Kang, H. Moon, Underground sewer pipe condition assessment based on convolutional neural networks, Automation in Construction 106 (2019) 102849, https://doi.org/10.1016/j.autcon.2019.102849.
[14] R.A. Fenner, Approaches to sewer maintenance: a review, Urban Water 2 (4) (2000) 343–356, https://doi.org/10.1002/1462-0758(200006)2:4<343::AID-UWAT5>3.0.CO;2-7.
[15] B. Hajek, Cooling schedules for optimal annealing, Math. Oper. Res. 13 (2) (1988) 311–329, https://doi.org/10.1287/moor.13.2.311.
[16] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Autom. Constr. 20 (8) (2011) 1211–1226, https://doi.org/10.1016/j.autcon.2011.05.005.
[17] J. Hacik, B.T. Adey, N. Lehan, Determination of near-optimal restoration programs for transportation networks following natural hazard events using simulated annealing, Comput. Aided. Civ. Infrastruct. Eng. 33 (8) (2018) 618–637, https://doi.org/10.1111/mice.12346.
[18] R. Navab-Kashani, F. Golkhou, Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition, J. Constr. Eng. Manag. 141 (6) (2015), 04015099, https://doi.org/10.1061/(ASCE)CE.1943-7862.0000976.
[19] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, in arXiv preprint arXiv:1804.02767, 2018, https://doi.org/10.1109/CVPR.2017.690.
[20] S. Kolender, D. Colloca, A. Boni, J. Kurash, Z. Seneviratne, State-of-the-art in sensor technologies for underground sewer inspections, Autom. Constr. 111 (2020) 103061, https://doi.org/10.1016/j.autcon.2019.102947.
[21] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Autom. Constr. 20 (8) (2011) 1211–1226, https://doi.org/10.1016/j.autcon.2011.05.005.
[22] R. Navab-Kashani, F. Golkhou, Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition, J. Constr. Eng. Manag. 141 (6) (2015), 04015099, https://doi.org/10.1061/(ASCE)CE.1943-7862.0000976.
[23] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, J. Kang, H. Moon, Underground sewer pipe condition assessment based on convolutional neural networks, Automation in Construction 106 (2019) 102849, https://doi.org/10.1016/j.autcon.2019.102849.
[24] R.A. Fenner, Approaches to sewer maintenance: a review, Urban Water 2 (4) (2000) 343–356, https://doi.org/10.1002/1462-0758(200006)2:4<343::AID-UWAT5>3.0.CO;2-7.
[25] B. Hajek, Cooling schedules for optimal annealing, Math. Oper. Res. 13 (2) (1988) 311–329, https://doi.org/10.1287/moor.13.2.311.
[26] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and contextual reasoning-based video interpretation method for rapid productivity analysis of construction operations, Autom. Constr. 20 (8) (2011) 1211–1226, https://doi.org/10.1016/j.autcon.2011.05.005.