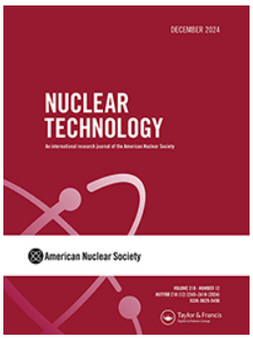


Nuclear Technology

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/unct20

Enhanced Video-Level Anomaly Feature Detection for Nuclear Power Plant Component Inspections Using the Latency Mechanism

Zhouxiang Fei, Callum Manning, Graeme M. West, Paul Murray & Gordon Dobie

To cite this article: Zhouxiang Fei, Callum Manning, Graeme M. West, Paul Murray & Gordon Dobie (2024) Enhanced Video-Level Anomaly Feature Detection for Nuclear Power Plant Component Inspections Using the Latency Mechanism, Nuclear Technology, 210:12, 2404-2418, DOI: 10.1080/00295450.2024.2337282

To link to this article: https://doi.org/10.1080/00295450.2024.2337282

Published online: 18 Jun 2024.

NUCLEAR TECHNOLOGY · VOLUME 210 · 2404–2418 · DECEMBER 2024

DOI: https://doi.org/10.1080/00295450.2024.2337282

Enhanced Video-Level Anomaly Feature Detection for Nuclear Power Plant Component Inspections Using the Latency Mechanism

Zhouxiang Fei,* Callum Manning, Graeme M. West, Paul Murray, and Gordon Dobie

University of Strathclyde, Department of Electronic and Electrical Engineering, Glasgow, United Kingdom

Received November 15, 2023
Accepted for Publication March 25, 2024

Abstract — *The conditions of various nuclear power plant facilities are regularly examined through manual inspections. Remote visual inspection is commonly applied and requires engineers to watch lengthy inspection footage and seek anomaly features therein. This is a labor-intensive process as anomaly features of interest usually only appear in very short segments of the original whole video. Therefore, an automated anomaly detection system is preferred to lessen the intensive labor cost in the inspection process. The detection process could also benefit from useful information that could potentially contribute to addressing reasoning traceability.*

With a well-prepared training data set of the anomaly feature, a convolutional neural network (CNN) can be developed to automatically detect anomaly indications in the inspection video. However, false-positive detections may occur and can be difficult to remove without seeking manual verification. To overcome this problem, we present a new automated video-level anomaly detection framework that utilizes the latency mechanism to effectively lessen false-positive occurrences, and therefore, increase detection accuracy. In this framework, a CNN-based anomaly classifier first performs initial scanning of the anomaly type of interest in every region of the sampled frames. Then our latency mechanism is applied to refine the initial scanning results by flagging up a region as an "anomaly" indication only when "anomaly" is detected by CNN in the current frame and also in a sequence of previous consecutive frames of the same region.

We present a case study of crack feature detection in superheater inspection videos to illustrate the performance of the proposed framework. The results show that the latency mechanism can effectively remove the original false-positive detections seen in the initial scanning. In order to provide a primary exploration of suggesting possible formats for addressing reasoning traceability, knowledge graphs of the reasoning process in the video-level detection framework are built to provide a better understanding of why a specific section of the video is flagged as anomaly contents by the video-level detection framework.

Keywords — *Crack detection, deep learning, knowledge graph, nuclear power plant inspection, remote visual inspection support.*

Note — *Some figures may be in color only in the electronic version.*

*E-mail: zhouxiang.fei@strath.ac.uk

I. INTRODUCTION

Inspections are routinely performed on a variety of nuclear power plant facilities to generate footage that can be analyzed to provide the status of the component structural health condition. The anomaly types of interest to be

identified may differ depending on the facility type, such as cracks in nuclear reactor cores[1–4] and corrosion in nuclear waste fuel storage canisters.[5] The understanding of facility conditions contributes to reviewing the defect growth and foreseeing the remaining life of the components so that essential maintenance operations can be subsequently scheduled.

One common approach to providing such understanding is through manual-based visual inspection and may involve reviewing real-time videos captured onsite depending on the inspection tools and procedures for the specific task. In spite of the procedural differences in conducting the inspection, one common challenge of the manual-based inspection is exposure to the excessively large volume of footage and the associated high level of constant and intensive concentration, resulting in a time-consuming and repetitive inspection process.

Taking crack feature detection in a superheater inspection[6] as background, a typical inspection video can last around 2 h, but may only contain short and intermittent crack feature video fragments of up to a few minutes. Therefore, manual inspection labor could be dramatically lessened if engineers only needed to review a summary of the suspected crack feature video segments (each lasting around 10 to 15 s) from the original footage and then make final informed decisions.

To realize this vision, autonomous anomaly feature detection is needed. In civil engineering, automated anomaly detection has been an intensive research field for assisting structural health monitoring of facilities with constraint access. For example, cracks in bridge deck surfaces,[7] steel beams,[8] and concrete[9] have been detected using image processing techniques. Recently, data-driven techniques have been widely utilized for automated anomaly detection. For example, the bag-of-visual-words approach was deployed to identify cracks on the graphite surface of a nuclear reactor core[3] and a classification system based on Haar-like features was trained to find cracks in wind turbine blades.[10]

Among a variety of data-driven techniques, the convolutional neural network (CNN) technique from the deep learning family draws increasing focus, as a CNN automatically learns the feature of interest for decision making from the training data set. For instance, the applications of CNN can be found in automated crack detection in railway tracks[11] and concrete structures at the patch level[12] and the semantic segmentation level.[13] Crack detection performance comparison between the data-driven-based CNN and edge detectors from the image processing field was investigated in Ref. [14].

However, we see limited applications of CNN in the nuclear power generation sector compared to the civil engineering field (e.g., pavement and bridge surface inspections). Possible factors in the nuclear sector may include the constraint availability of anomaly types and the rigorous regulation of the traceability of automated decision making and the use of synthetic data. Currently, some application examples of CNN for anomaly inspection in nuclear facilities can be seen in crack detection in reactor surface imitations,[15] crack-like feature detection in pressure vessels (e.g., superheaters[16]), and corrosion identification in nuclear fuel waste storage canisters.[5]

Note that the aforementioned research frameworks[5,11,12,15,16] mainly delved into accurate anomaly detection at the image level (or referred to as the frame level if extracted from inspection videos). Challenges remain in advancing the anomaly inspection from the frame level to accurate and efficient video-level detection. Specifically, the main purpose of video-level detection is to identify the video fragments containing anomaly indications in the original entire video. However, many current works have solely focused on further enhancing anomaly detection accuracy in single images, though spatiotemporal information of an anomaly feature in consecutive frames could be utilized to improve detection accuracy in single frames.[15] Supported by video-level detection, inspection engineers only need to review and validate a collection of video fragments containing anomaly features, instead of watching the entire length of the original video.

Furthermore, false-positive detections at the frame level may occasionally occur in autonomous anomaly detection. For instance, the strong edge features of tube holes in superheaters may be mistakenly detected as crack features. Such false-positive detections can be difficult to correct using only single-frame analysis. By nature, video-level detection is useful for effectively removing false-positive detections using the prior information in consecutive frames.

On this basis, we first introduce a new video-level detection framework in this paper to facilitate the accurate and efficient capture of anomaly feature video fragments and the effective removal of false-positive detections. Under this framework, the frames extracted from the inspection video are first sampled intermittently, and each sampled frame is split into full-grid nonoverlapping regions. Initial scanning is then performed using a trained CNN of the anomaly type to check for anomaly indications in each region of the sampled frames. Next, the proposed latency mechanism is applied to rectify the false-positive detections in the initial scanning results.

During the latency mechanism process, a region in a sampled frame is flagged as "anomaly" only when the following two conditions are satisfied: (1) anomaly indications are identified in this region of the current frame and (2) the number of previous consecutive frames with anomaly indication detected at the same region is equal to or above a customized threshold. As a result, the false-positive decisions made in the initial scanning can be effectively rectified using this latency mechanism.

In practice, blurry frames from the video may not produce meaningful detection information and could cause interruption to the continuity of the frame content categories in adjacent frames. To solve this problem, we apply a sliding window technique after the latency mechanism process to continuously register the status of each frame as "anomaly feature" or "nonanomaly." As a result, continuous anomaly feature fragments can be accurately detected from the remote visual inspection footage of nuclear power plant components.

Furthermore, due to the multiple parameters used in this decision-making system, it is beneficial to explore possible approaches to provide improvement in result traceability. This is because traceability is one of the key factors for addressing the trust and safety requirements of the highly regulated nuclear industry. A knowledge graph[17] could be a potentially suitable approach for investigating its application for providing traceability of how the video-level detection system produces a specific conclusion.

This motivates the second aspect of this paper, which is to deploy knowledge graphs to investigate the application of knowledge graph visualization for addressing result traceability. In this way, such an investigation attempts to provide useful information to address the reasoning traceability requirement to some extent. For our video-level detection framework, knowledge graphs are designed to capture useful information in each step of the reasoning pipeline, from frame sampling to sliding window smoothing operations. We can then query and visualize the knowledge graph to obtain a better understanding of why a certain video segment is flagged as an anomaly by the video-level detection framework.

Overall, this paper first demonstrates a novel application of the proposed video-level detection framework for the inspection videos of superheaters (a type of steam cycle component in nuclear power plants), and then applies knowledge graphs to explore their potential application for contributing to the result traceability for the video-level detection process. Other inspection scenarios in pipes and vessels on nuclear sites could also potentially benefit from the proposed approach.

We organize the remainder of this paper as follows. Section II details our proposed video-level anomaly feature detection methodology and provides a concise introduction to knowledge graphs. An application case study of superheater inspection videos is presented in Sec. III with results discussed. Section IV presents an illustration of the knowledge graph application for our detection framework and discusses how the presented knowledge graph visualization could be used to provide better understanding of reasoning traceability in our detection framework. Section V summarizes the conclusions and outlines future work.

II. METHODOLOGY

The development of the proposed anomaly detection framework involves two main stages: (1) developing an anomaly feature classifier and (2) applying the classifier in the video-level detection process. A variety of data-driven techniques (such as in Refs. [15,18]) can be successfully applied to design an anomaly feature classifier. We deploy the deep learning technique in this paper.

II.A. Concise CNN Introduction

The deep learning technique family has been popular for developing autonomous anomaly detection systems based on representative training data sets and accurate training performance. Generally speaking, in a fundamental CNN structure, convolution layers are first deployed and followed by pooling layers to extract the anomaly feature of interest from the image. Then the fully connected and softmax layers are applied to obtain the classification result. Inside the convolution layer, filters are convolved with different sections of the image to produce feature maps. The feature maps are then processed by the pooling layer to reduce the feature map size and rearranged into an array as the input to the fully connected layer. The softmax layer turns the output of the fully connected layer into the scores of each category. The content in the image is classified as the category with the highest score. A detailed introduction to the CNN technique can be found in Ref. [19].

Preparing the training data set plays a vital role in developing a CNN classifier, as the correct features of interest can be learned for classification. However, manual labeling of data sets can be a labor-intensive process due to the large number of labeled images required for each category. To reduce the manual work of preparing data sets, we propose an automated labeling technique[6]

that is versatile for efficiently producing labeled images of a variety of anomaly features. Our research work in Ref. [6] has been advanced in this paper by performing video-level anomaly inspection via a trained CNN classifier. The GoogLeNet architecture, originally proposed in Ref. [20] to classify 1000 different object categories in ImageNet,[21] was chosen as the backbone of the CNN classifier in the scanning stage of the video-level classification process in Fig. 1. The development process of our GoogLeNet-based classifier is outlined in Fig. 1a and detailed in Ref. [16].

II.B. Video-Level Detection

II.B.1. Frame Sampling

Video-level detection involves consecutive frame-level detections and the analysis of the frame-level detection results. A full-grid scanning strategy is applied with a CNN classifier trained for the anomaly type to perform frame-level detection. Taking crack features in superheaters as the anomaly type, the CNN classifier takes in a patch with a resolution of 224 × 224 × 3 in RGB (red, green, and blue) format and classifies the patch content as crack feature or noncrack class. Taking the frame-level detection illustration in Fig. 2 as an example, each patch region in the full grid is first scanned using the CNN classifier. The patches classified as crack feature are flagged as potential crack regions to be confirmed with the latency mechanism.

Performing video-level detection by analyzing all the frames in the video can be computationally intensive. In practice, adjacent frames typically share a large level of redundant information. For this reason, we introduce frame sampling as the first step in the video-level detection workflow in Fig. 1b to reduce computation cost and accelerate the process. In this paper, frame sampling rate is defined as the index interval between two analyzed frames. Taking a frame sampling rate of 3 as an example, only the first, fourth, seventh, and 11th (and so forth) frames are analyzed via the full-grid scanning by the classifier.

II.B.2. Latency Mechanism

After the full-grid scanning of the sampled frames, we apply the latency mechanism to the initial scanning results, as outlined in Fig. 1b. The latency mechanism leverages the prior information about the nature of a video inspection (as opposed to a single frame) to enhance detection performance. Specifically, the false-positive classifications in the initial frame-level classification results can be effectively filtered out using a latency threshold value.

We register a patch region in a sampled frame as anomaly only if the CNN classifier detects anomaly feature in the same region across the current and previous (latency threshold value–1) sampled frames. For example, let us assume the index of the superheater inspection frame in Fig. 2a is 1000th in the video, with frame sampling rate = 3 and latency threshold value = 3. The patch at (location index: 2) the 1000th frame is registered as crack feature only when the CNN detects crack feature at the patch region (index: 2) in the 994th, 997th, and 1000th frames. One assumption of the latency mechanism is that the anomaly features remain in the same patch region during the short

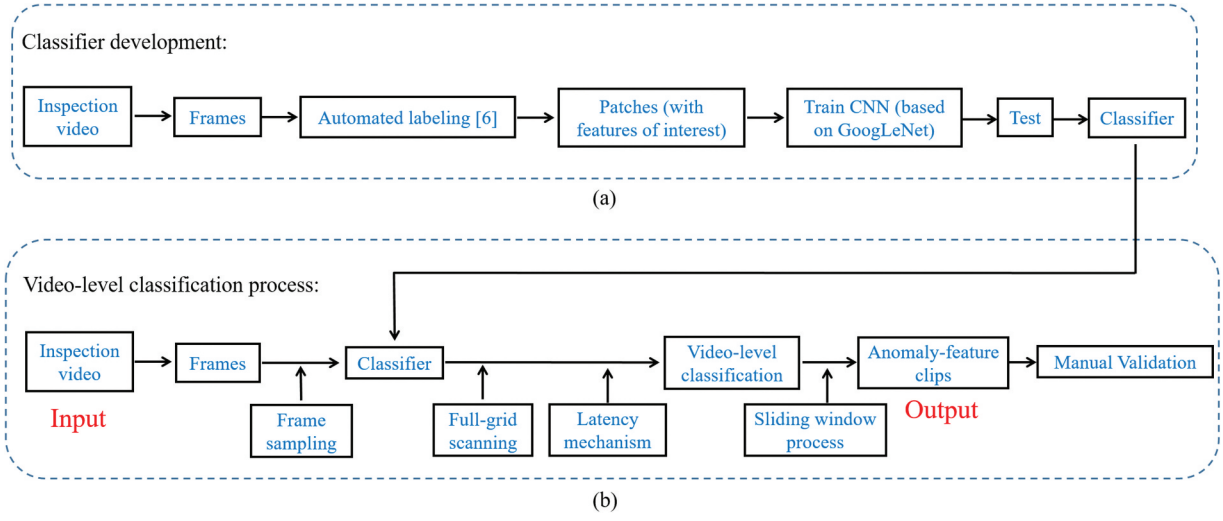


Fig. 1. Overall framework: (a) development of the CNN classifier and (b) video-level detection process.

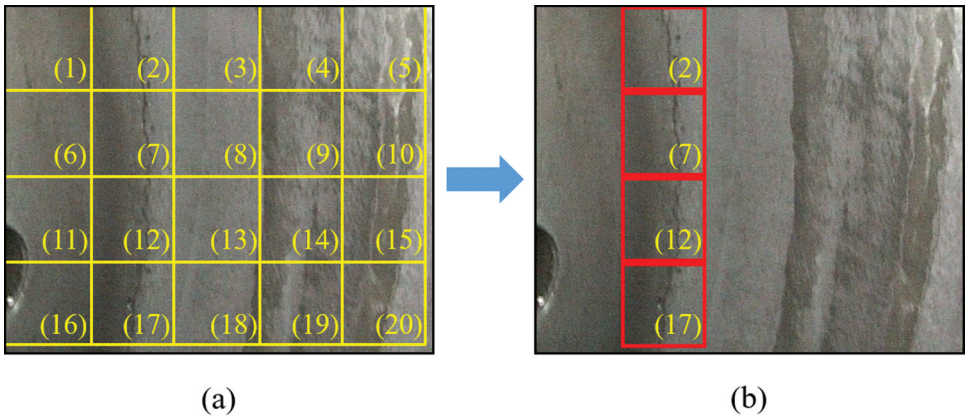


Fig. 2. Full-grid anomaly scanning strategy illustration. The patch footer represents its region index in the frame. (a) Example of full-grid scanning for crack features at different locations (patch size: 224 × 224 × 3 in RGB channels) in a superheater inspection frame and (b) classified patches as potential crack feature regions by the CNN.

latency time window. This assumption is usually true in most cases, but may require customized tuning for a suitable latency threshold value for a specific task.

In general, the moving speed of the inspection camera influences the choice of the latency threshold value to ensure that the potential anomaly feature stays in the target patch region during the latency period. For example, a large latency threshold value corresponds to a long time window and may omit the registering of anomaly features that only last for a very short duration in the inspection footage. Now we register each initially scanned frame as a potential anomaly frame if any patch region in this frame is registered as an anomaly feature by the latency mechanism; otherwise this frame is registered as a potential normal frame.

II.B.3. Sliding Window Process

Note that the video-level detection result is still discrete after the latency mechanism process. This is because the initial scanning of some blurry frames may not produce meaningful information of the frame contents. The blurriness in the frames could be induced by swift camera movement and dark lighting conditions. As a result, we need to apply a sliding window technique to smooth out the discontinuity in the frame-level detection results and obtain continuous anomaly feature video segments for engineers to review.

In this step, we apply a sliding window with a specific window duration (referred to as the sliding window length) to go through the frames processed by the latency mechanism. If the number of analyzed frames registered as potential anomaly in the sliding window is larger than or equal to a user-defined sliding window threshold, we perform the following two steps: (1) all the frames (both analyzed frames and unsampled frames in between) in this window interval are registered as anomaly and (2) a 3-s period is attached to the beginning and end of the sliding window to attempt the capture of camera moving toward and away from the anomaly region.

This step may generate a very large number of short anomaly feature clips. In practice, it is highly plausible that neighboring anomaly clips may refer to the same anomaly indication. Therefore, the adjacent anomaly feature clips (including the gap in between) apart by less than 10s are merged to sensibly reduce the number of summary clips for engineers to cross validate. In total, there are four tunable parameters used in the video-level anomaly detection framework, listed as frame sampling rate, latency threshold value, sliding window length, and sliding window threshold.

II.C. Knowledge Graph Concise Background

Knowledge graphs are a useful means of representing information due to their inherent flexibility and applicability across diverse domains.[22,23] Fundamentally, a knowledge graph is comprised of entities, portrayed as nodes, interconnected through relationships, depicted as edges. This straightforward graph structure offers a robust and intuitive framework for organizing and presenting information. The key advantages to employing knowledge graphs as a medium for storing and reasoning about the data in a system are as follows.

First, knowledge graphs can be used as general knowledge bases. Their flexible structure means a wide variety of domain knowledge can be encoded without

needing rigid schema. This feature makes knowledge graphs ideal for consolidating distributed data into one unified data resource that can be queried.[24]

Second, knowledge graphs can be used to facilitate complex inferences by incorporating the embedded information in the knowledge graph. By analyzing how the entities within the knowledge graph are related, complex deductive reasoning can be performed.

Furthermore, knowledge graphs could be a potentially promising approach for contributing to addressing the need for explainable artificial intelligence (AI). This is because knowledge graphs allow for capturing expert knowledge, which could be interpreted eventually in a human-readable and machine-legible way. This feature could drive the promising progress to reveal the AI reasoning process. This potential benefit could particularly coincide with addressing the strict requirement of AI decision-making traceability in the nuclear industry sector.

In our study, we used the reasoning rules of the decision-making process in Sec. II.B to build the knowledge graphs of the video-level detection process. Specifically, the knowledge graphs were used to store and visualize the status of each patch location in every sampled frame along the video-level detection pipeline. The results, discussed in Sec. IV, investigate the potential benefit of knowledge graph application in revealing result traceability.

III. CASE STUDY: SUPERHEATER CRACK FEATURE DETECTION

III.A. Superheater Inspection Introduction

We demonstrate the proposed video-level detection methodology through a case study of superheater inspection. A superheater is a thermal component used in the steam cycle to overheat steam in nuclear power plants. Remote visual inspection is routinely conducted to ensure the safe continued operation of superheaters. In the inspection process, an endoscope is sent into the superheater through the access point to examine the existence and growth of crack-like features in the bottom circumferential area (also referred to as the tube plate upper radius) of the superheater.

An inspection video is generated from this process and reviewed to record the identified observations of crack features. Figure 3 presents a schematic of the superheater structure in the context of this paper. Our aim is to automatically identify the short clips of crack features from the original video through the proposed video-level anomaly detection workflow without missing crack feature observations. Therefore, engineers only need to review a small number of short summary clips instead of going through the entire inspection video, thus lightening the associated manual labor cost.

III.B. Case Study Results

A CNN classifier based on the GoogLeNet structure[20] was developed, with the data sets prepared via the automated labeling technique in Ref. [6]. The training, validation, and testing data sets accounted for 60.5%, 15.5%, and 24% of the total 800 crack feature patches and 800 noncrack patches, respectively. Each data set was balanced, and there was no data leakage between any data sets (i.e., the data from a specific video was solely in one of the three data sets). A training process of 30 epochs via transfer learning[25] resulted in a precision of 94.59%, a recall of 91.15%, a F1-score of 92.84%, and a total accuracy of 92.97% for the testing data set. A detailed introduction of this CNN-based crack feature detector for superheater inspection can be found in Ref. [16].

Figure 4 demonstrates the performance of the latency mechanism on removing false-positive detections in a testing video. Specifically, the frames in the testing video were extracted and full-grid scanned with the CNN classifier at frame sampling rate = 1 (i.e., every frame was analyzed). Figures 4a through 4d and Figs. 4i through 4l present the initial scanning results, while Figs. 4e through 4h and Figs. 4m through 4p are the corresponding counterparts processed by the latency mechanism at latency threshold value = 3. As can be seen, the rich textural features in the frames (due to discoloration, welding, etc.) could trick the CNN classifier into making false-positive detections, which can be challenging to rectify without utilizing the prior information between neighboring frames. However, such false-positive detections can be effectively reduced using the latency mechanism.

Comparing the counterparts between Figs. 4a and 4e, Figs. 4i and 4m, and Figs. 4b and 4f, we can see that the false-positive detections due to intensive discoloration of edge features and tube hole edges are precisely removed through the latency mechanism while preserving the true-positive detections. Figures 4j and 4n, Figs. 4c and 4g, Figs. 4k and 4o, and Figs. 4d and 4h provide further examples of removing false-positive detections triggered by tube hole contours, welding, strong lighting reflection, and outlet contours, respectively. Note that the false-positive detections made in blurry frames can also be

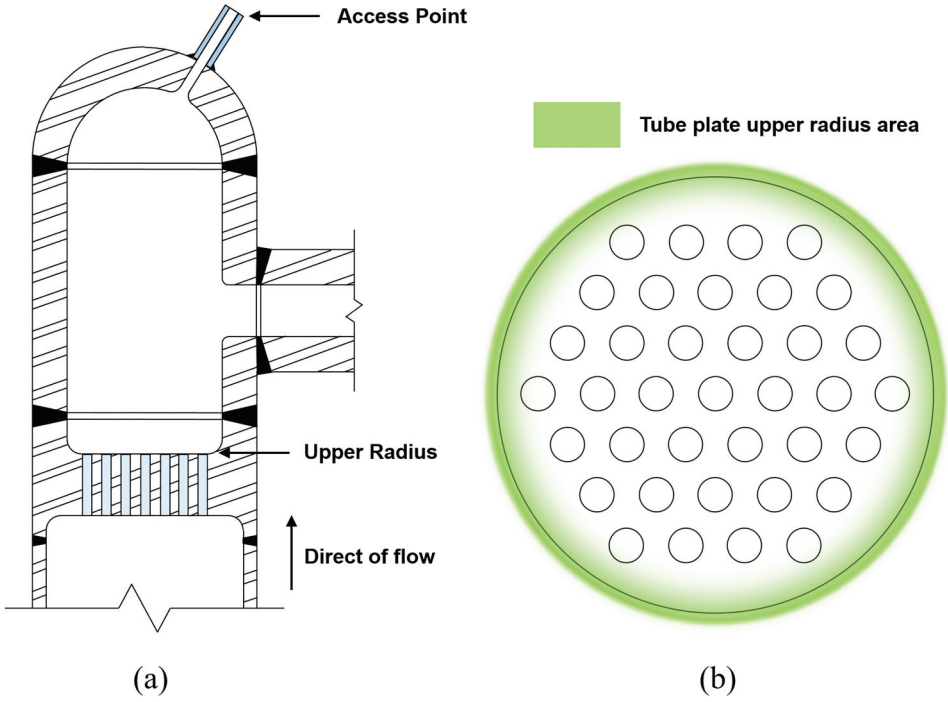


Fig. 3. Superheater schematic: (a) cross-section view and (b) top view of the tube plate.

removed through the latency mechanism, as exemplified in Figs. 4l and 4p.

The detected crack feature patch number before and after the latency process in each frame of the testing video (comprising 5551 frames recorded at 25 frames/s) is shown in Fig. 5 and matched against the ground-truth contents at different times. Manual review of the video contents was performed to obtain the ground-truth starting and ending frame indices of the noncrack and crack feature segments. Figure 5a shows the number of crack feature patches detected only using the initial full-grid scanning. As can be seen, genuine crack features in the anomaly sections of the video are accurately detected. Although false-positive detections occur at the noncrack sections, the number of these misclassifications in the frames within the noncrack sections is generally small.

As shown in Fig. 5b, detection performance was further enhanced using the latency mechanism with latency threshold value = 3. Comparing Fig. 5a with Fig. 5b, the number of frames with false-positive detections (highlighted by green boundaries) is reduced by 91% (from 376 to 35). Therefore, it is clear that the latency mechanism is effective in filtering out the false-positive detections in the noncrack sections, thus improving detection performance. Note that the frame blurriness may obstruct the decision making of the CNN classifier, resulting in no regions of interest flagged in some blurry frames.

Next, we performed the sliding window process introduced in Sec. II.B.3 to obtain crack feature summary clips with the final registration status of each frame. Figure 6 shows the crack feature summary clips obtained using the proposed video-level detection framework at different parametric settings. As can be seen in Fig. 6a, the anomaly summary clips obtained at a frame sampling rate of 3, a latency threshold value of 2, a sliding window length of 15, and a sliding window threshold of 6 are a good match with the ground-truth crack feature segments in the video, as all the recorded crack features in the official inspection report have been successfully captured.

There is a discrepancy in the duration between the automatically generated summary clips and the corresponding ground-truth anomaly sections in Fig. 6a due to the blurriness in some frames caused by the swift endoscope movements. The parametric setting in Fig. 6a was used as the benchmark to illustrate the effect of parametric tuning on video-level detection, as shown in Figs. 6b through 6e. By comparing between Figs. 6a and 6b and between Figs. 6a and 6c, it can be seen that reducing the sliding window threshold or increasing the sliding window length relaxes the criteria of flagging a crack feature segment.

Such parametric tuning is more inclined to trigger false-positive detections, such as the false-positive detections at the start of the testing video in Fig. 6b and between the crack feature sections (i.e., 2 and 3) in Figs. 6b and 6c. On the other hand, as can be seen by comparing Figs. 6a and 6d, increasing the latency threshold value could toughen the criteria of flagging a crack feature section, resulting in the

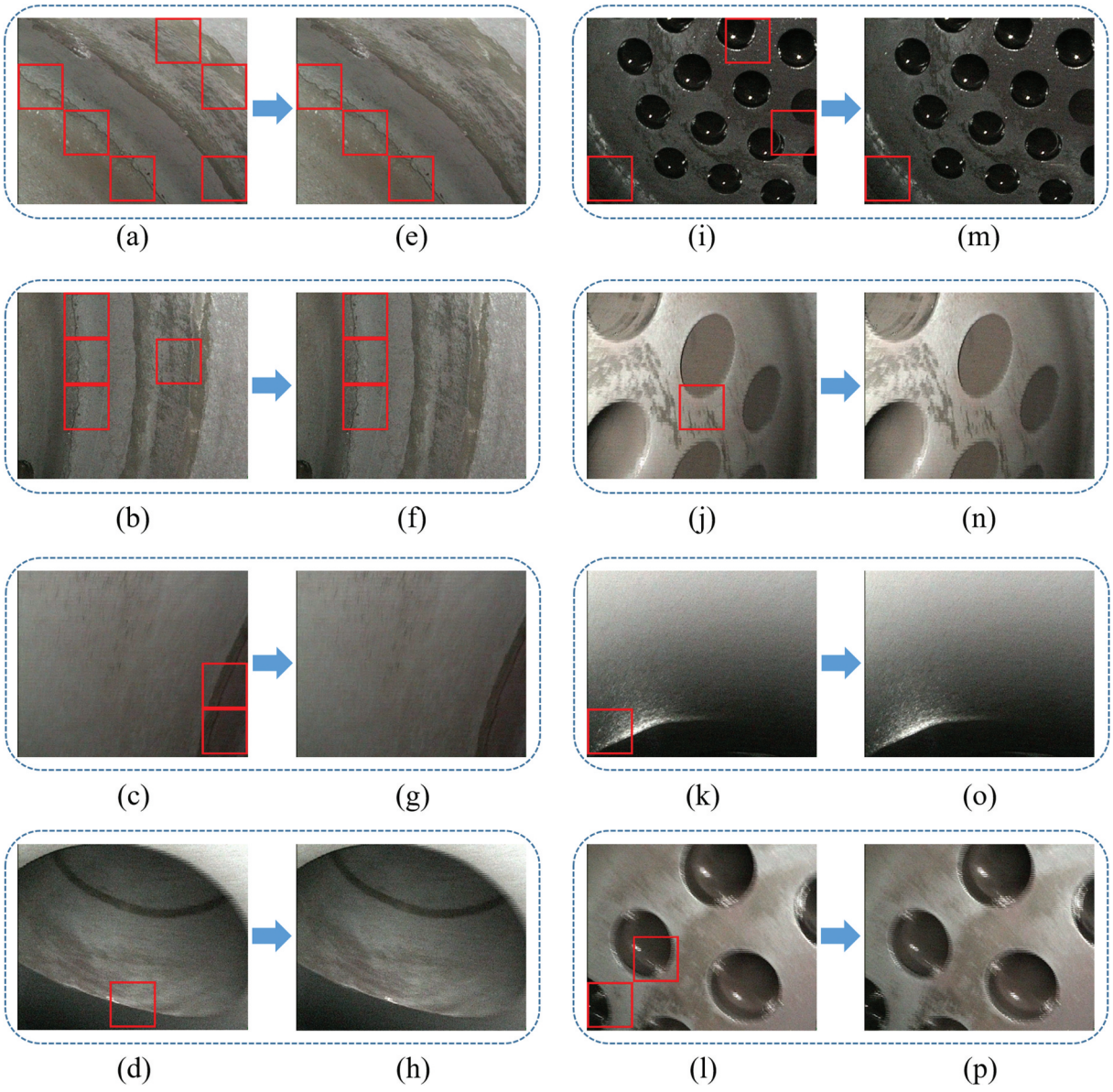


Fig. 4. Examples of rectified false-positive detections via the latency mechanism with frame sampling rate = 1 and latency threshold value = 3. Figs. 4a through 4d and 4i through 4l demonstrate the false-positive detections in the initial full-grid scanning stage. These false-positive detections were mainly attributed to the strong textural features due to discoloration, contours of tube holes and outlets, stains, and light reflection off metallic surfaces. In Figs. 4e through 4h and 4m through 4p, the efficacy of the latency mechanism is demonstrated for removing such false-positive detections.

crack feature segment (i.e., 2) being missed in the detection result in Fig. 6d. By comparing Figs. 6a and 6e, it can be seen that increasing the frame sampling rate could dilute the detection of crack features, resulting in the crack feature segment (i.e., 2) being omitted from the detection result in Fig. 6e.

It is noted that finding the optimal parametric setting in the four-dimensional space (formed by the frame sampling rate, latency threshold value, sliding window length, and sliding window threshold) is a computationally challenging task that is fundamentally associated with the characteristics of the analyzed video. The adoption of the video-level inspection framework may require customized parametric tuning by other researchers for their own contexts of anomaly detection.

III.C. Discussion

Taking the superheater crack feature inspection as the anomaly detection context, the proposed video-level

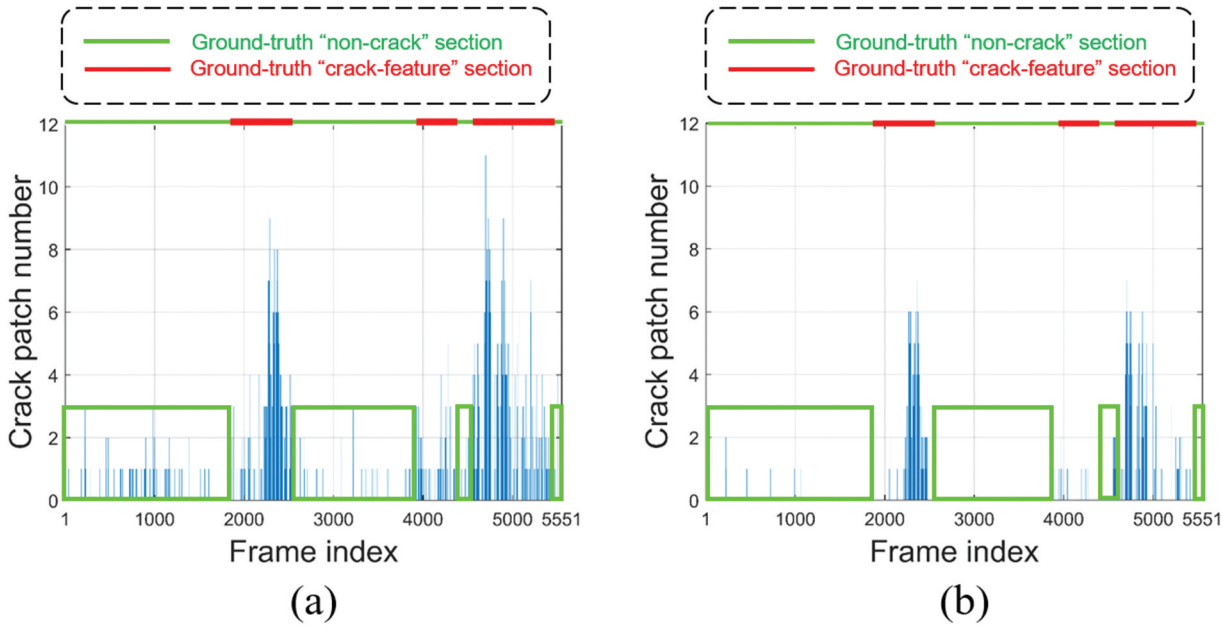


Fig. 5. Performance of the latency mechanism on filtering out false-positive detections in the testing video at frame sampling rate = 1: (a) detected crack feature patch number in each frame only using the initial full-grid scanning and (b) refined classification results using the latency mechanism with a latency threshold value of 3.

detection framework has been demonstrated to successfully, accurately, and efficiently identity anomaly content segments in the inspection video. Specifically, removing false-positive detections can be challenging using only single-frame analysis. However, our proposed latency mechanism, based on the prior information in neighboring frames, has been shown to be effective in filtering out false-positive detections.

On this basis, the sliding window process was applied to convert discrete frame-level detection results into consecutive anomaly summary clips for specialists to cross validate. The proposed framework is general and straightforwardly transferrable to a variety of anomaly inspection scenarios in nuclear power plants. Choosing parameter values in the proposed framework is dependent on the specific inspection context, and therefore requires customized tuning for the task of interest. With the effective removal of false-positive detections addressed in this paper, one important direction of future work is the development of a technique for rectifying false-negative detections. Future work will also focus on tracking the movement of anomaly features across neighboring frames to further improve anomaly detection accuracy.

It is worth noting that while the efficacy of our video-level inspection framework was demonstrated using the GoogLeNet model as the anomaly feature detector, the same framework could, in theory, be applied to other deep learning–based systems that are used for video-level anomaly (e.g., crack) detection. This is because in our framework, the CNN-based classifier (in this study, GoogLeNet) for anomaly detection is an interchangeable component and can be replaced with other well-trained CNN models for various inspection tasks. Potential candidates for deployment include but can go beyond ResNet,[26] VGG,[27] and AlexNet.[28] Please refer to Ref. [29] for a comparative study of different state-of-the-art CNN models on concrete surface crack detection evaluated for the trade-off between inference time and accuracy. A comprehensive comparison study of different CNN models is beyond the scope of this paper.

IV. KNOWLEDGE GRAPH RESULTS

IV.A. Knowledge Graph Implementation

In the context of this paper, traceability refers to the demonstration of the processed results at each step of the video-level detection workflow in a hierarchical visualization format. Note that the definition of traceability can take a variety of forms in different contexts. The discussion of traceability in this paper only focuses on the status of patches, frames, and window intervals at each step in the proposed decision-making process.

In our study, the knowledge graphs were used to describe pairs of two entity names and the relationship

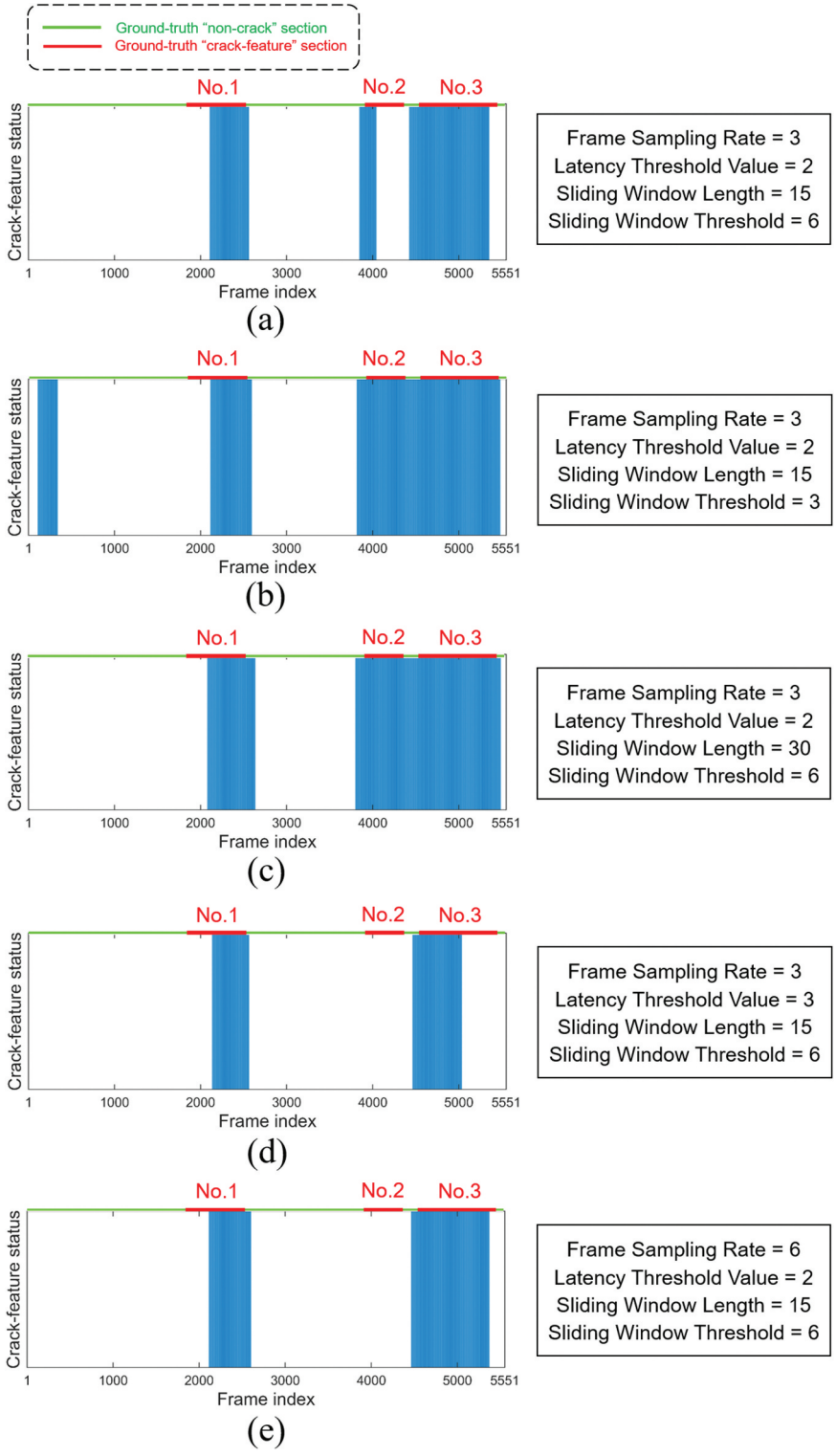


Fig. 6. Automatically generated crack feature summary clips using the proposed video-level detection methodology at different parametric settings in contrast to the ground-truth anomaly segments (referred to as 1, 2, and 3) in the video. Note that the conditional 3-s extension at each side of the sliding window and the 10-s automated grouping rule from Sec. II.B.3 were performed to give the detection results.

in between. Our knowledge graph implementation was primarily based on the Graph objects provided in the MATLAB environment. Specifically, the nodes of the Graph object were used to represent (1) the entities (e.g., Patches, Frames, Window intervals) in the video-level detection system, and (2) the attributes (i.e., Crack feature, Flagged, and Anomaly) that describe those entities. The edges of the Graph object represent the relationships (e.g., Next, Previous, in) between those various nodes. The rules of the underlying mechanism in the video-level detection process were used to encode the relationships in the knowledge graph.

As the knowledge graph follows the same reasoning process of the video-level detection framework, building the knowledge graph requires information from the CNN's evaluation of each patch in every sampled frame, as well as the parameter values (i.e., frame sampling rate, latency threshold value, sliding window length, and sliding window threshold) used in the video-level detection process. The schema[30] that was used to build the knowledge graphs is illustrated in Fig. 7.

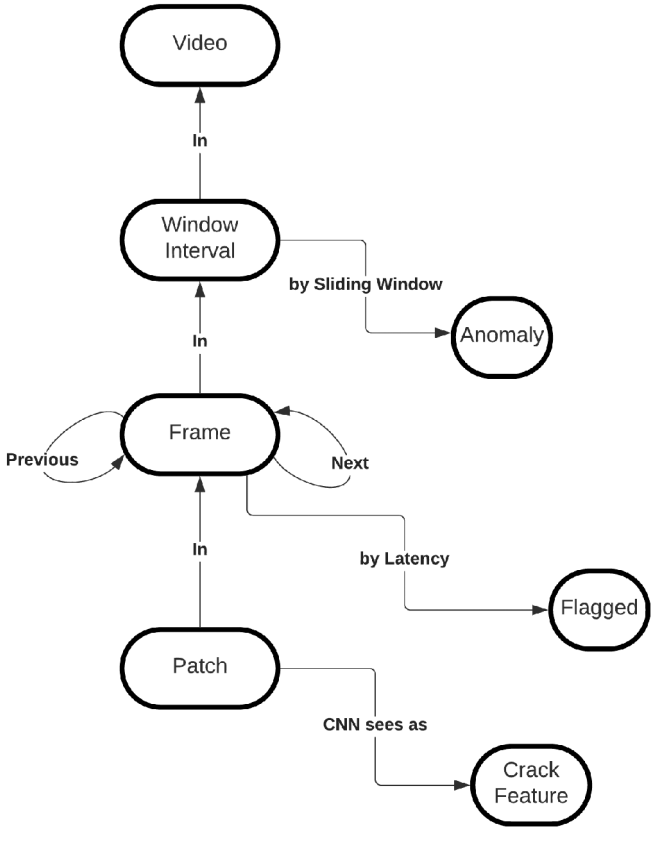


Fig. 7. Schema of the knowledge graph for the video-level detection framework to illustrate the underlying structure of how the nodes and edges are related to each other.

The schema in Fig. 7 is described as follows. The entity nodes are Patch, Frame, Window interval, and Video. The attribute nodes required are Crack feature, Flagged, and Anomaly. The edges in the schema are in, Next, Previous, by Latency, by Sliding Window, and CNN sees as.

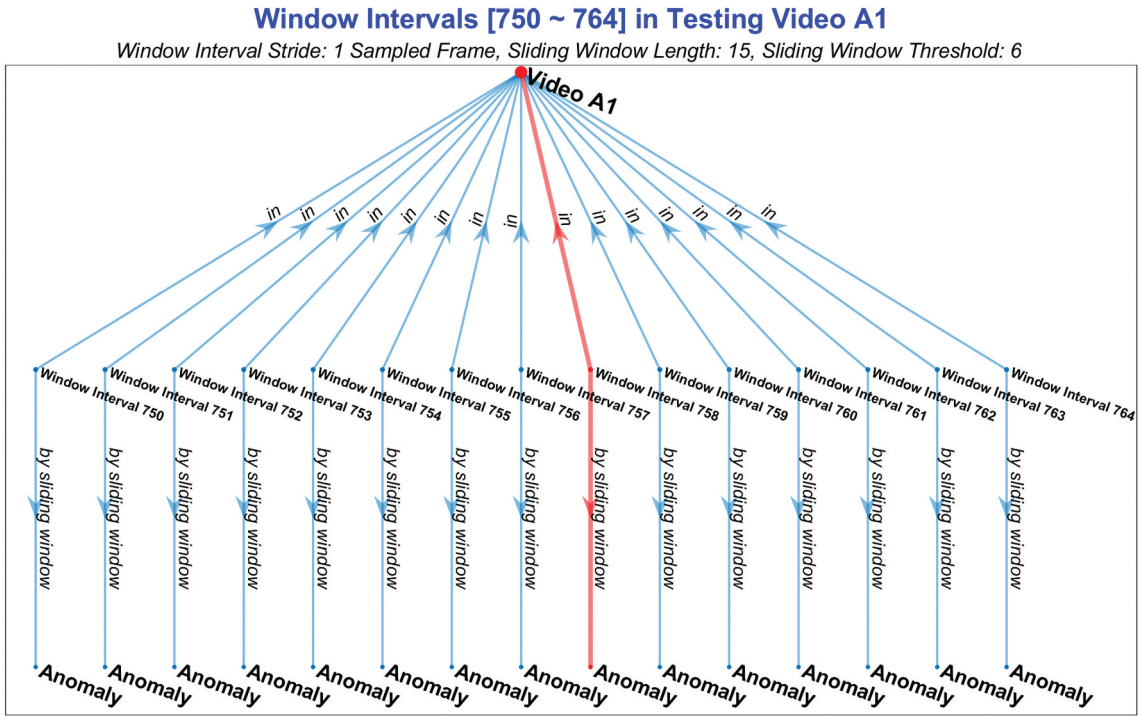
The nodes and edges interact in the following way. The Patch node is connected to the Frame node through the edge in, and the same principle applies between the Frame and Window interval nodes and between the Window interval and Video nodes. A Patch entity node is connected to a Crack feature attribute node via the CNN sees as edge if the patch is detected as a potential crack feature in the initial full-grid scanning process.

A Frame entity node is connected to a Flagged attribute node through a by Latency edge if any patch in the frame is determined to be a crack feature by the latency mechanism. A Window interval entity node is connected to an Anomaly attribute node through the by Sliding Window edge if this Window interval satisfies the sliding window threshold. Each Frame node is connected to the previous and next Frame nodes via the Previous and Next edges, respectively (if adjacent frames exist).

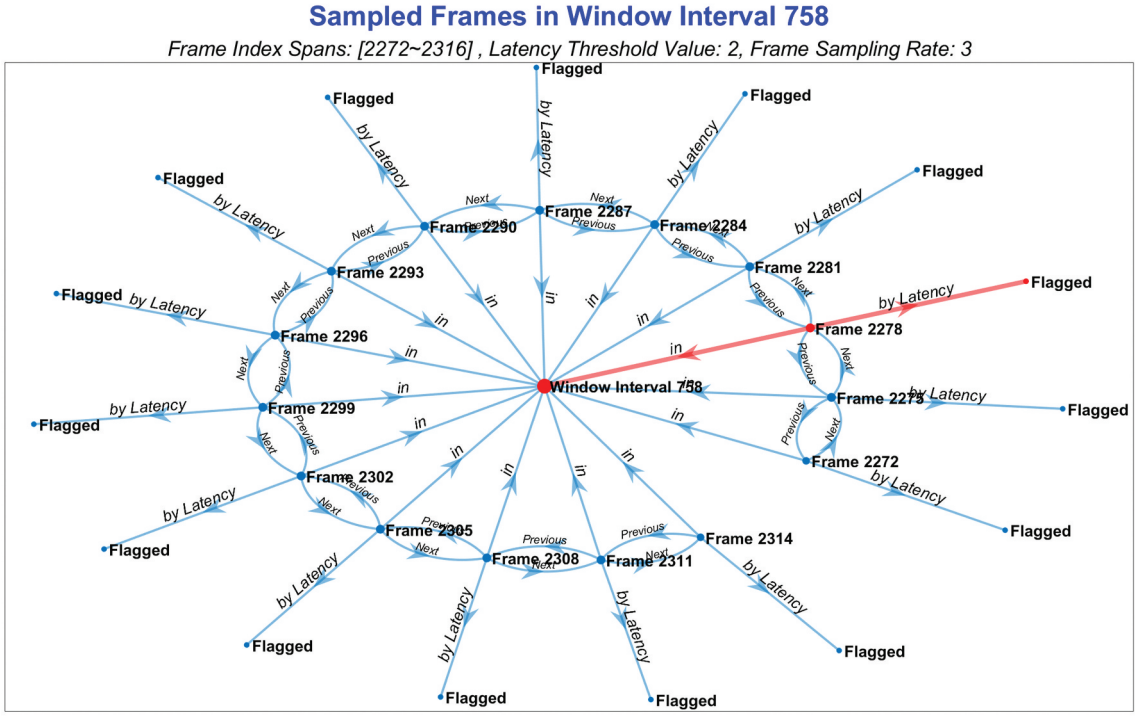
The rules for building the knowledge graph mirrors the underlying reasoning mechanism of the video-level detection system. To start with, we query if a sampled Frame node is connected to any Patch node which is in turn connected to a Crack feature attribute node. If so, this sampled Frame node is referred to as the original starting Frame, and we note the region index of the Patch node(s) associated with the Crack feature attribute. We then move to the previous sampled Frame node and repeat the query to check if there is a Patch node(s) with the Crack feature attribute at the same region(s) as in the original starting Frame.

This process is repeated recursively until the latency threshold value is reached or the latency condition does not suffice. If the latency threshold value is reached, we then connect the original starting Frame node to a Flagged attribute node via a by Latency edge to log that this Frame node contains a crack feature. We can then query a Window interval node (with the same length as the sliding window) to see if it comprises any sampled Frame node with the Flagged attribute. If the number of sampled Frame nodes with Flagged attribute satisfies the sliding window threshold, then this Window interval node is connected to an Anomaly attribute node via a by Sliding Window edge.

Based on this schema definition, the knowledge graphs of the testing video results (video referred to as A1, the same as the one used in Sec. III.B) are successfully constructed. Figure 8 shows only a subsection of the knowledge graphs for brevity. As can be seen, the

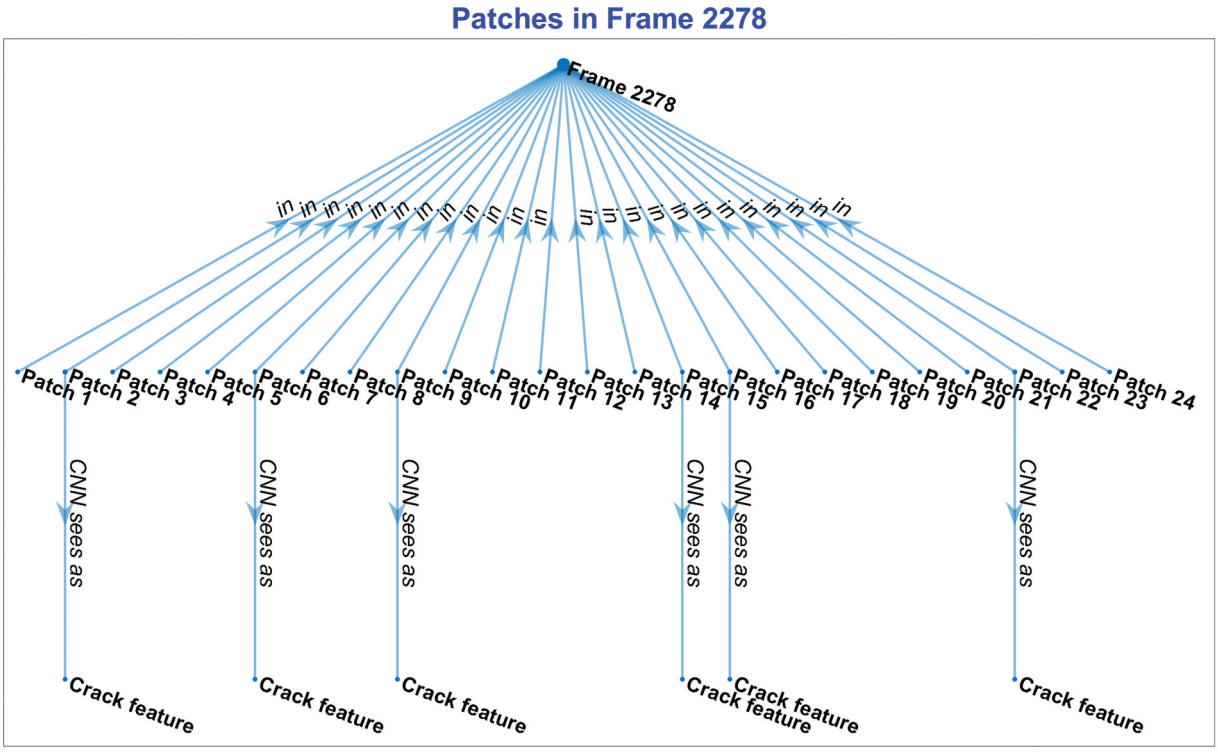


(a)



(b)

Fig. 8. Knowledge graph visualization to illustrate the status of the Window interval, Frame, and Patch nodes in an exemplar anomaly segment of the testing video A1 outlined via a three-stage process: (a) the Video-to–Window interval stage where only a short selection of Window interval nodes are presented herein for legibility; (b) the Window interval–to-Frame stage where each Window interval node contains 45 (i.e., frame sampling rate × sliding window length) Frame nodes and only the sampled Frame nodes are provided for brevity; and (c) the Frame-to-Patch stage where the scannable area in each frame consists of 24 patch regions.



(c)

Fig. 8. (Continued).

knowledge graphs present the status of all the entity nodes at each step in the video-level detection workflow in order to provide useful information for addressing the traceability of the underlying decision-making process. Specifically, Fig. 8a shows a subsection of Window interval nodes (ranging from Window interval 750 to Window interval 764) that are connected to the Anomaly attribute nodes based on the rules of the detection framework. Taking the highlighted Window interval 758 branch (between the 2272nd and 2316th frames) as an example, this Window interval is flagged as Anomaly, as the number of sampled Frame nodes herein with the Flagged attribute satisfies the sliding window threshold.

The status of the sampled Frame nodes in Window interval 758 are given in Fig. 8b. Finally, taking the sampled Frame 2278 node branch as an example, Fig. 8c shows the status of the Patch nodes in the Frame 2278 node that are seen by CNN as crack feature. Note that the conditional 3-s extension at each side of the sliding window and the 10-s automated grouping rule were not used in Fig. 8. As can be seen in this example, the useful information from the knowledge graph visualization can help provide a better understanding of why a specific Window interval is logged as Anomaly, through the status illustration of the associated Patch and Frame nodes in each step of the video-level detection pipeline.

IV.B. Knowledge Graph Discussion

Note that we only applied the knowledge graph visualization to provide a better understanding of why a specific section of the inspection video was flagged as anomaly by the video-level detection framework. There was no attempt to address adding clarity to the deep learning (i.e., CNN) classification part, which only acts as the first step to performing initial scanning in the decision-making process.

V. CONCLUSIONS

This paper introduced a new video-level detection framework capable of accurately and efficiently identifying anomaly feature contents in inspection videos. The

support from this framework can dramatically reduce the intensive manual labor cost of the inspection task and assist engineers with informed decision making. The proposed workflow was general and considered applicable to a variety of video-based inspection scenarios in nuclear power plants. We used the crack-like feature inspection in superheaters as a scenario example to demonstrate the performance of our detection workflow.

The results showed that the proposed framework can precisely detect the genuine crack features while effectively filtering out false-positive detections through the latency mechanism. On this basis, short crack feature summary clips were accurately obtained by applying the sliding window technique over the filtered discrete frame-level detection results.

Furthermore, knowledge graph visualization of the results at each step of the decision-making process was successfully implemented and illustrated based on the same underlying reasoning mechanism. The useful information from knowledge graph visualization was used to provide a better understanding of why a specific segment of the inspection video was detected as anomaly by the video-level detection framework.

Future work will partly focus on developing techniques to rectify false-negative detections and further enhance anomaly detection through the use of movement vector knowledge across neighboring video frames. Future work will also further explore the potential benefit of applying the knowledge graph to provide a common language that is human readable and machine legible so that traceability of the decision-making process can be addressed.

Disclosure Statement

No potential conflict of interest was reported by the authors.

Funding

This work was carried out as part of a collaborative research project funded by the Advanced Nuclear Research Centre at the University of Strathclyde and supported by Altrad Babcock Ltd.

ORCID

Zhouxiang Fei http://orcid.org/0000-0002-5003-3949
Callum Manning http://orcid.org/0009-0001-5293-5022
Graeme M. West http://orcid.org/0000-0003-0884-6070
Paul Murray http://orcid.org/0000-0002-6980-9276
Gordon Dobie http://orcid.org/0000-0003-3972-5917

References

1. G. WEST et al., "Improved Visual Inspection of Advanced Gas-Cooled Reactor Fuel Channels," *Int. J. Progn. Health Manage.*, **6**, *3*, 1 (2015).
2. P. MURRAY et al., "Automated In-Core Image Generation from Video to Aid Visual Inspection of Nuclear Power Plant Cores," *Nucl. Eng. Des.*, **300**, 57 (2016); http://dx.doi.org/10.1016/j.nucengdes.2015.11.037.
3. M. DEVEREUX, P. MURRAY, and G. WEST, "A New Approach for Crack Detection and Sizing in Nuclear Reactor Cores," *Nucl. Eng. Des.*, **359**, 1 (2019).
4. M. DEVEREUX, P. MURRAY, and G. WEST, "Automated Object Detection for Visual Inspection of Nuclear Reactor Cores," *Nucl. Technol.*, **208**, *1*, 115 (2022); http://dx.doi.org/10.1080/00295450.2020.1863067.
5. T. PAPAMARKOU et al., "Automated Detection of Corrosion in Used Nuclear Fuel Dry Storage Canisters Using Residual Neural Networks," *Nucl. Eng. Technol.*, **53**, *2*, 657 (2021); http://dx.doi.org/10.1016/j.net.2020.07.020.
6. Z. FEI et al., "Automated Generation of Training Dataset for Crack Detection in Nuclear Power Plant Components," *Proc. 12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technology*, June 14–17, 2021, p. 251.
7. I. ABDEL-QADER, O. ABUDAYYEH, and M. E. KELLY, "Analysis of Edge-Detection Techniques for Crack Identification in Bridges," *J. Comput. Civil Eng.*, **17**, *4*, 255 (2003); http://dx.doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255).
8. C. M. YEUM and S. J. DYKE, "Vision-Based Automated Crack Detection for Bridge Inspection," *Comput.-Aided Civ. Infrastruct. Eng.*, **30**, 759 (2015); http://dx.doi.org/10.1111/mice.12141.
9. T. YAMAGUCHI et al., "Image-Based Crack Detection for Real Concrete Surfaces," *Trans. Electr. Electron. Eng.*, **3**, 128 (2008); http://dx.doi.org/10.1002/tee.20244.
10. L. WANG and Z. ZHANG, "Automatic Detection of Wind Turbine Blade Surface Cracks Based on UAV-Taken Images," *IEEE Trans. Ind. Electron.*, **64**, *9*, 7293 (2017); http://dx.doi.org/10.1109/TIE.2017.2682037.
11. D. LI et al., "Acoustic Emission Wave Classification for Rail Crack Monitoring Based on Synchrosqueezed Wavelet Transform and Multi-Branch Convolutional Neural

Network," *Struct. Health Monit.*, **20**, *4*, 1563 (2021); http://dx.doi.org/10.1177/1475921720922797.

12. Y. J. CHA, W. CHOI, and O. BÜYÜKÖZTÜRK, "Deep Learning–Based Crack Damage Detection Using Convolutional Neural Networks," *Comput.-Aided Civ. Infrastruct. Eng.*, **32**, *5*, 361 (2017); http://dx.doi.org/10.1111/mice.12263.
13. L. ZHANG, J. SHEN, and B. ZHU, "A Research on an Improved Unet-Based Concrete Crack Detection Algorithm," *Struct. Health Monit.*, **20**, *4*, 1864 (2021); http://dx.doi.org/10.1177/1475921720940068.
14. S. DORAFSHAN, R. J. THOMAS, and M. MAGUIRE, "Comparison of Deep Convolutional Neural Networks and Edge Detectors for Image-Based Crack Detection in Concrete," *Constr. Build. Mater.*, **186**, 1031 (2018); http://dx.doi.org/10.1016/j.conbuildmat.2018.08.011.
15. F. C. CHEN and M. R. JAHANSHAHI, "NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion," *IEEE Trans. Ind. Electron.*, **65**, *5*, 4392 (2018); http://dx.doi.org/10.1109/TIE.2017.2764844.
16. Z. FEI et al., "CNN-Based Automated Approach to Crack-Feature Detection in Steam Cycle Components," *Int. J. Press. Vessels Pip.*, **207**, 1 (2024); http://dx.doi.org/10.1016/j.ijpvp.2023.105112.
17. C. GUTIERREZ and J. F. SEQUEDA, "Knowledge Graphs," *Commun. ACM*, **64**, *3*, 96 (2021); http://dx.doi.org/10.1145/3418294.
18. M. R. JAHANSHAHI and S. F. MASRI, "Adaptive Vision-Based Crack Detection Using 3D Scene Reconstruction for Condition Assessment of Structures," *Autom. Constr.*, **22**, 567 (2012); http://dx.doi.org/10.1016/j.autcon.2011.11.018.
19. C. AGGARWAL, *Neural Networks and Deep Learning: A Textbook*, 1st ed., Springer, New York (2018).
20. C. SZEGEDY et al., "Going Deeper with Convolutions," *Proc. 2015 IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, Massachusetts, June 7–12, 2015, p. 1.
21. J. DENG et al., "ImageNet: A Large-Scale Hierarchical Image Database," *Proc. 2009 IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, Florida, June 20–25, 2009, p. 248.
22. Y. NING et al., "UUKG: Unified Urban Knowledge Graph Dataset for Urban Spatiotemporal Prediction," *Proc. 37th Conf. on Neural Information Processing Systems*, New Orleans, Louisiana, 2023, p. 1.
23. E. RAJABI and S. KAFAIE, "Knowledge Graphs and Explainable AI in Healthcare," *Information*, **13**, *10*, 1 (2022); http://dx.doi.org/10.3390/info13100459.
24. A. HOGAN et al., *Knowledge Graphs*, 1st ed., Springer, Switzerland (2022).
25. K. GOPALAKRISHNAN et al., "Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-Based Data-Driven Pavement Distress Detection," *Constr. Build. Mater.*, **157**, 322 (2017); http://dx.doi.org/10.1016/j.conbuildmat.2017.09.110.
26. K. HE et al., "Deep Residual Learning for Image Recognition," *Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, June 27–30, 2016, p. 770.
27. K. SIMONYAN and A. ZISSERMAN, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Proc. 3rd Int. Conf. on Learning Representations*, San Diego, California, May 7–9, 2015, p. 1.
28. A. KRIZHEVSKY, I. SUTSKEVER, and G. E. HINTON, "Imagenet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, **60**, 84 (2017); http://dx.doi.org/10.1145/3065386.
29. A. S. RAO et al., "Vision-Based Automated Crack Detection Using Convolutional Neural Networks for Condition Assessment of Infrastructure," *Struct. Health Monit.*, **20**, 2124 (2021); http://dx.doi.org/10.1177/1475921720965445.
30. A. FAGAN, G. WEST, and S. MCARTHUR, "Leveraging Knowledge from Historic Engineering Drawings," *Proc. 13th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technology*, Knoxville, Tennessee, July 15–20, 2023, p. 1.