The Journal of Supercomputing (2024) 80:19374–19394
https://doi.org/10.1007/s11227-024-06185-8

Quantifying response latency in video surveillance systems using object detection techniques

Jia Miao[1] · Li Zhu[1] · Hongli Zhao[1] · Sen Lin[1] · Xinjun Gao[2]

Accepted: 3 May 2024 / Published online: 26 May 2024

Abstract

As video surveillance systems become increasingly essential for railway operations, accurate and precise performance testing is crucial. Traditional methods for response latency testing rely on manual readings with millisecond-level clocks, which can lead to compatibility issues, software crashes, and potential security risks. To address these challenges, this paper proposes a response latency testing method based on object detection for railway video surveillance systems. The response latency test method includes two application scenarios: real-time video call and pan–tilt–zoom camera control response. By leveraging the YOLO-V5 model and object detection techniques, the response speed of railway video surveillance systems is effectively evaluated, ensuring testing precision. Experimental results validate the efficiency and feasibility of the proposed approach, emphasizing its enhanced stability and compatibility compared to traditional methods. The proposed approach offers an innovative solution for testing the response lantency of railway video surveillance systems, contributing to the enhancement and optimization of railway operations.

Keywords Object detection · Video surveillance systems · Response latency · Control response

Abbreviations

PTZ	Pan–tilt–zoom
YOLO	You only look once
NMS	Non-maximum suppression
ORB	Oriented FAST and Rotated BRIEF

✉ Hongli Zhao
hlzhao@bjtu.edu.cn

[1] State Key Lab of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, People's Republic of China

[2] Communications Signal Research Institute of Railway Research Institute, Beijing, People's Republic of China

FAST Features from accelerated segment test
BRIEF Binary robust independent elementary features
BF Brute-force

1 Introduction

With the rapid development of railway transportation, there is widespread concern about the safety and efficiency of train operations [1–5]. The railway video surveillance system has been widely applied as a crucial means to ensure railway security [6]. The railway video surveillance system enables the timely detection and early warning of railway accidents and safety hazards, while also improving railway operational efficiency. However, performance issues have also emerged with the railway video surveillance system, which can directly impact the system's operational efficiency and data processing capabilities. These performance issues include video capture, image quality, system reliability, and system response latency [7–9]. To meet the growing demands for video surveillance performance, assessing and improving the performance indicators of the video surveillance system have become potential solutions [10]. Considering the diverse nature of performance indicators, achieving rapid and accurate performance evaluation of the video surveillance system has become a primary concern.

The complexity and physical limitations of video transmission make the introduction of video playback latency inevitable [11]. Currently, there are latency assessment methods tailored for various analog and digital camera models [12]. Moreover, to conduct user-centered video latency testing, [11] proposes a method for measuring glass-to-glass video latency using video conferencing as an example. This approach involves using the VideoLat testing tool, which needs to be deployed on both the frontend and backend. It also requires calibration measurements before conducting latency tests to ensure the accuracy of the results. With the rapid development of cloud-based real-time intelligent video systems, accurately measuring end-to-end latency has become a focal point of interest among scholars. [13] proposes three methods to measure latency: timecode, remote online, and lossless remote video online.

Considering the challenges of remote video transmission and multiple transmission nodes in railway video surveillance systems, especially in large stations where multiple frontend devices are installed, the hardware deployment approach mentioned above for response latency testing is not feasible in practical engineering testing.

Over the past few years of integration and testing, the team has made heavy use of tools such as screenshots, screen recording, and millisecond clock detection. However, these tools are feature-rich and predominantly green software, which results in frequent installation and uninstallation during testing. In actual testing, it was found that the adopted tools have varying degrees of compatibility issues with systems from different vendors. Furthermore, software crashes or automatic disconnections frequently happen during use, directly impacting testing progress and quality.

Currently, when considering the network security of railway systems, the utilization of green software may pose risks such as system virus intrusion.

Therefore, while implementing the response latency test of the video surveillance system on the existing railway video surveillance system platform, solving the compatibility of test tools from different manufacturers has become the core challenge at present.

The main contributions of this paper can be summarized as follows:

- **We propose an innovative approach to develop a response latency measurement system that utilizes object detection technology.** This method solves compatibility issues that may arise when combining different detection tools and effectively addresses challenges related to manual response latency calculation. The measurement system integrates the functions of various detection tools and directly displays and stores the measured response latency results.
- **We utilize the YOLO-V5 model**[14] **to monitor the start time** (t_s) **of a precisely controlled video surveillance system.** The YOLO-V5 model detects the halo that appears when the mouse clicks the pan–tilt–zoom (PTZ) camera rotation button. Once the halo indicating the button click is detected, the current time is recorded as t_s.
- **In the real-time video call scenario, we calculate the sum of the differences of the mean values of the three channels of the image RGB to determine the end time** (t_e). The first moment when the successful loading of the video is detected is considered as t_e.
- **In the PTZ camera control response scenario, we utilize a combination of sign matching and contour detection to identify the end time** (t_e). When a successful PTZ response is detected, the system logs the initial moment when the object area changes beyond a predefined threshold, which signifies the end time (t_e) latency of the PTZ camera response.

The following article will be structured as follows. In the Related Work section, we will discuss the current video response latency measurement techniques and the specific requirements of the railway field for the response latency of two scenarios: real-time video call and PTZ camera control response. The Methods Used for Response Latency Measurements section explains in detail the methods used in the response latency measurement process. Section 4 will present the architecture of our proposed response latency measurement method for object detection-based video surveillance systems. Next, Sect. 5 will evaluate the effectiveness of our proposed response latency measurement method using offline operational videos. Finally, Sect. 6 will summarize and draw conclusions based on the previous sections.

2 Related work

In this section, we provide a brief overview of previous research focusing on surveillance latency measurements, as well as describe the methodology currently used by the Railway Joint Coordination Group to measure the response latency of video

surveillance systems. Finally, we outline the specific requirements for response latency in two scenarios: real-time video call and PTZ camera control response, in the technical specifications of the Chinese Railway.

2.1 Video response latency measurements

The camera has a response latency between capturing the scene and outputting the video signal. In recent years, only a few new methods have emerged for measuring surveillance latency, primarily focusing on camera response latency and real-time video loading latency. Sven Ubik et al. [15] proposed three methods for measuring camera response latency: timecode views, waveform offsets, and on-screen photodetectors. However, these methods require additional equipment and are not practical for engineering measurements with limited resources. On the other hand, live streaming or video buffering latencies are usually assessed using sophisticated test scripts or test systems designed for measuring response latency, as outlined in references [16, 17]. In [18], a method is presented for measuring the response and processing latency of the PTZ camera with an AI engine, but this may necessitate additional device requirements for deploying the AI engine.

In the railway video surveillance system, the test computer used by the integrated video joint commissioning team is quite old. As a result, its response speed has gradually decreased over time. To test the control response latency of a video surveillance system, testers typically use a combination of simple tools. For instance, they open a millisecond clock display tool next to the client's PTZ camera video surveillance screen and utilize recording software to capture the operator's PTZ camera control commands and the camera's corresponding response.

To measure the response latency of a video surveillance system, the tester records the start time (t_s) when the operator issues the control command and the first moment (t_e) when the client's video surveillance screen changes. The response latency is then calculated by subtracting t_s from t_e. However, the use of multiple tools during the testing process can often cause compatibility issues, resulting in inaccurate data.

Moreover, the manual reading method of response latency measurement is prone to human error, as different operators may produce inconsistent results during measurement. This can lead to inaccurate data, especially when dealing with large-scale data, which may require a lot of time and human resources to measure. This is not practical for real-time monitoring systems that require immediate results.

2.2 Response latency measurement requirements for two railway video scenes

As per the Q/CR 575-2022 *Technical Specification for Railway Integrated Video Surveillance System*[19] issued by the China National Railway Group, the railway integrated video surveillance system is classified into a four-tier architecture consisting of front-end equipment, video nodes, video terminals, external systems, and network equipment that connect these four parts through switches or communication devices. The system structure diagram is shown in Fig. 1.

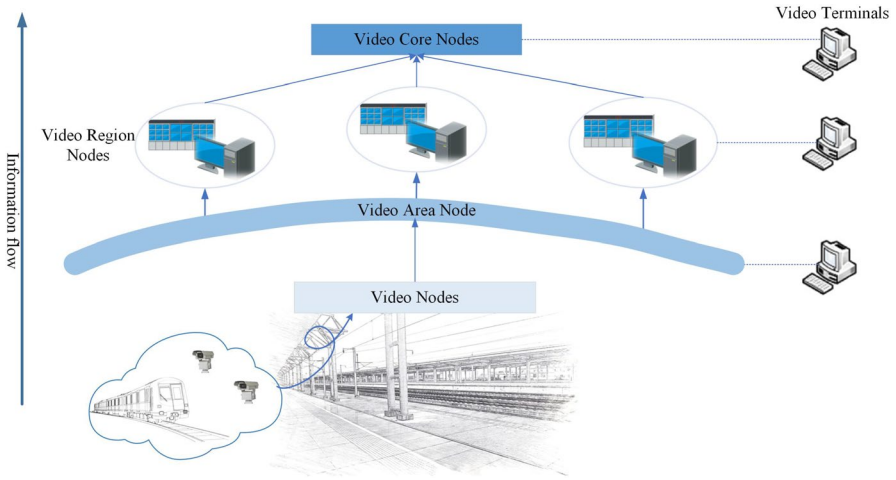

Fig. 1 Architecture diagram of railway video surveillance system. The architecture diagram includes the components, and how they are interconnected to enable video surveillance and security management of the railway area

Response latency is defined as the time latency between image capture at a video node and its display at a video endpoint. Minimizing response latency is crucial for enabling real-time monitoring and effective decision-making. Performance testing must carefully measure video call latency and PTZ camera control response latency to ensure timely and accurate data. The new specification states that system response latency requirements dictate that real-time video call latency should be limited to 3 s. Additionally, the response latency for PTZ camera control should not exceed 500 milliseconds.

In conclusion, comprehensive performance testing that meets these requirements enables railway operators to accurately evaluate the effectiveness, reliability, and robustness of their video surveillance systems. This, in turn, ensures that these systems can operate optimally, thereby significantly contributing to the security and efficiency of the railway network.

3 Methods used for response latency measurements

In this study, we propose an innovative response latency testing methodology based on object detection techniques. This methodology is specifically designed to evaluate response latency metrics for two scenarios: real-time video call and PTZ camera control response. The method relies on object-detection and involves several stages. A comprehensive schematic of the testing process is shown in Fig. 2.

In the following sections, we will explain the key techniques used in this response latency test methodology, in order to better understand its underlying mechanisms.

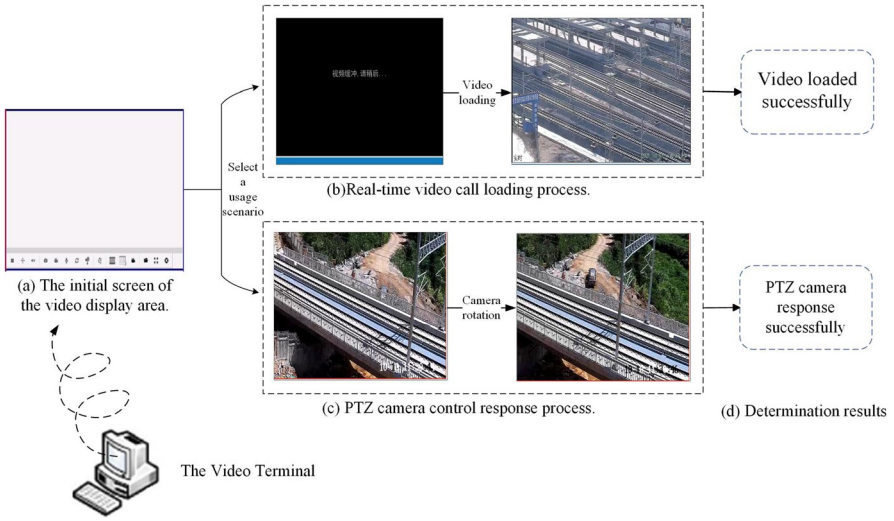

Fig. 2 Comprehensive schematic diagram of the testing process. The change process of the control response latency measurement interface for two scenarios of real-time video call and PTZ camera response

3.1 YOLO-V5 model

YOLO (You Only Look Once) [20] is a state-of-the-art object detection system that operates in real-time. The YOLO-V5 model, a remarkable advancement in real-time object detection, serves as a robust and efficient framework for object detection tasks. In our proposed performance testing method, this model plays a crucial role in the object detection process.

To facilitate subsequent image processing, the input images are typically resized to a standardized size. The input image pixels of our network are set to [640, 640]. Subsequently, the images undergo a series of convolutional neural network layers for feature extraction, capturing features such as edges, textures, and shapes. After feature extraction, YOLO-V5 utilizes detection heads to perform object detection on the images. The detection heads consist of a set of convolutional layers responsible for predicting bounding boxes and corresponding class probabilities of the objects. YOLO-V5 adopts anchor boxes to assist in predicting objects of various sizes and aspect ratios. As the same object may be detected by multiple anchor boxes, non-maximum suppression (NMS) is utilized to eliminate overlapping bounding boxes, ensuring the retention of the most accurate object detection results. Ultimately, YOLO-V5 outputs the predicted object bounding boxes and corresponding class probabilities, along with the final detection results after NMS.

In the context of our performance testing method, YOLO-V5 is used to detect when a user initiates a video call or activates PTZ camera control, which then triggers the time latency timer. The model's real-time detection capability enables it to

quickly and accurately identify these initiations, thereby ensuring the precision of the subsequent response latency measurement.

3.2 ORB algorithm-based feature matching

In computer vision, feature matching algorithms are essential for tasks such as object detection and image alignment. The Oriented Fast and Rotated Brief (ORB) algorithm is particularly useful in real-time applications due to its rapid feature extraction and optimized matching strategy [21, 22]. The algorithm's effectiveness stems from its integration of the Features from Accelerated Segment Test (FAST) keypoint detector and the Binary Robust Independent Elementary Features (BRIEF) descriptor generator, which together facilitate a comprehensive image characterization and matching process.

Mathematically, the ORB algorithm can be described as a two-stage process: keypoint detection and keypoint description.

The keypoint detection stage employs the FAST algorithm, which operates by analyzing the luminance of a central pixel and its surrounding pixels within a predefined circular neighborhood. The FAST algorithm evaluates the difference in intensity between the central pixel and a threshold number of contiguous pixels arranged in a specific pattern. Mathematically, for a pixel p at coordinates (x, y) and a set of pixels N within the neighborhood, the FAST algorithm computes:

$$I(p) - \tau \geq \sum_{q \in N} w_q (I(q) - I(p)) \tag{1}$$

where $I(p)$ is the intensity of the central pixel, τ is a predetermined threshold, and w_q represents the weights assigned to the pixels in N. If this condition is met, the central pixel is considered a potential keypoint.

To enhance the detection process, the ORB algorithm incorporates orientation assignment, which aligns keypoints with local image gradients. The orientation θ of a keypoint is determined by the dominant direction of gradients in its vicinity:

$$\theta = \arg\min_{\phi} \sum_{(x,y) \in N} w_{x,y} \sin(\phi - \angle G_{x,y}) \tag{2}$$

where $G_{x,y}$ represents the gradient vector at pixel location (x, y), and $w_{x,y}$ is the gradient magnitude-weighted value.

The keypoint description stage leverages the BRIEF algorithm to generate binary descriptors. For each keypoint, the BRIEF algorithm selects a set of n pixel pairs within a defined radius and computes the grayscale differences between them. The resulting differences are then thresholded to generate a binary string, which forms the descriptor. Mathematically, let D_i be the difference in intensity between the i-th pixel pair (x_i, y_i) and (x_{i+1}, y_{i+1}), and let T be a predetermined threshold. The binary descriptor b is constructed as follows:

$$b_i = \begin{cases} 0 \text{ if } D_i \leq T, \\ 1 \text{ otherwise.} \end{cases} \tag{3}$$

For $i = 1, 2, \ldots, n$. This binary descriptor is compact, efficient to compute, and exhibits robustness against various image transformations and noise.

By combining these stages, the ORB algorithm achieves a balance between speed, accuracy, and robustness, making it a popular choice for feature matching in real-time computer vision systems. The use of binary descriptors, as opposed to real-valued descriptors, further reduces storage requirements and computational complexity, while maintaining a high level of matching performance [23].

3.3 Homography matrix-based camera movement distance calculation

To eliminate the impact of PTZ camera jitter on camera rotation detection, we utilize feature matching in conjunction with a single homography matrix [24]. This helps us calculate the movement distance of objects in the front and back video frames.

In image processing, the homography matrix is a crucial mathematical tool. Monoclinicity or projection transformation is a type of transformation that maps points from one plane to another. It has a wide range of applications in image processing, such as correcting perspective distortion, creating panoramic images, and inferring relative motion between two images.

In the context of distance detection for PTZ camera images, the homography matrix quantifies the transformation between the initial and resultant states of the PTZ camera image. This calculation helps measure the distance covered by the video image as a result of PTZ camera control commands.

Assume a point $p_1 = (x_1, y_1, 1)^{\mathrm{T}}$ in the first image has a corresponding point $p_2 = (x_2, y_2, 1)^{\mathrm{T}}$ in the second image. The homographic transformation relationship between the two images can be represented as follows:

$$p_2 \sim H p_1 \tag{4}$$

Where "∼" represents the two vectors are in proportion, and H is the homography matrix. Because p_2 and p_1 are both homogeneous coordinates, H is a homography matrix, mathematically, that is expressed as follows:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{5}$$

By substituting the coordinates of p_1 and p_2 into the above equation, we can get:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{6}$$

In the computation of the homography matrix H, the homogeneous coordinate system is employed, and h_{33} is set to 1 to ensure the matrix's uniqueness. After rearranging the terms, the subsequent expressions are derived:

$$x_2 = \frac{x_1 h_{11} + y_1 h_{12} + h_{13}}{x_1 h_{31} + y_1 h_{32} + 1} \tag{7}$$

$$y_2 = \frac{x_1 h_{21} + y_1 h_{22} + h_{23}}{x_1 h_{31} + y_1 h_{32} + 1} \tag{8}$$

By leveraging feature point matching and least squares estimation techniques, a precise homography matrix H is obtained, which encapsulates the transformation relationship between images. This matrix is then used to calculate the horizontal and vertical displacements of the images, thereby recognizing image movement. If the calculated displacement exceeds a pre-defined threshold, it indicates a successful PTZ camera response.

The robustness of this approach is attributed to its capability to accommodate both rotational and translational camera movements, making it a compelling choice for detecting PTZ control responses with high precision and reliability. When combined with the YOLO-V5-based object detection model outlined in the previous subsection, it forms a comprehensive system for measuring and evaluating PTZ camera response time within railway video surveillance systems.

4 Object detection-based performance testing method for railway video surveillance systems

During the response latency testing of a video surveillance system, we need to collect two important pieces of information—the start time (t_s) and the end time (t_e). To determine the start time (t_s), we use the YOLO-V5 model for mouse click aperture detection. Additionally, we utilize image processing and object detection methods to capture the moment when the video loads successfully or the PTZ camera responds successfully.

4.1 YOLO-V5-based mouse call operation detection

To ensure the accurate recording of the control start times (t_s) of the video surveillance system, YOLO-V5 is utilized in this paper. It monitors and recognizes the clicking actions of testers who use the mouse to issue control commands. The start time is when the tester initiates various functions by clicking on the control buttons in the interface of the video surveillance system.

The YOLO-V5 model divides the image into grids and assigns object detection tasks to each grid cell. The test image demonstrates this process, as detailed

in Fig. 4. Each cell predicts multiple bounding boxes and the class probabilities of those boxes. The bounding boxes are weighted according to the predicted probabilities. Object detection is done in a single pass, making YOLO-V5 faster than other object detection methods.

In the response latency test method for video surveillance systems, YOLO-V5 is utilized to detect when the tester initiates issuing control commands. This triggers the clock to accurately record the start time (t_s). The model's real-time detection capability enables quick and accurate detection of the onset of a mouse click. This ensures the accuracy of subsequent response latency measurements.

Algorithm 1 Workflow for Determining Screen Image Changes

Input: RGB three-channel average value of the previous frame: pre_r, pre_g, pre_b.
Output: Does the surveillance screen change? (Boolean output).

for each frame in the video stream **do**
  if test time less than 5s **then**
    Get the RGB three-channel average of the current video frame: cur_r, cur_g, cur_b.
    Calculate the sum of differences: $b = |cur_r - pre_r| + |cur_g - pre_g| + |cur_b - pre_b|$.
    Compare b with the threshold value θ.
    if $b < \theta$ **then**
      Add True to the sequence M.
    else if $b \geq \theta$ **then**
      Add False to the sequence M.
    end if
    Update the previous frame's RGB average: $pre_r = cur_r, pre_g = cur_g, pre_b = cur_b$.
  end if
end for

4.2 Real-time video call latency measurement method

Digital images, represented as matrices within computing systems, encode essential characteristics such as brightness, color, and additional image information, as depicted in Fig. 3. In the context of railway surveillance videos, digital image processing proves instrumental. For the latency measurement of real-time video calling in the video surveillance system, the end time is measured using the algorithm 1.

We utilize the OpenCV [25, 26] to analyze video objects, aiming to simplify the reading and processing of video files. This approach enables the processing of images on a frame-by-frame basis across the entire video file.

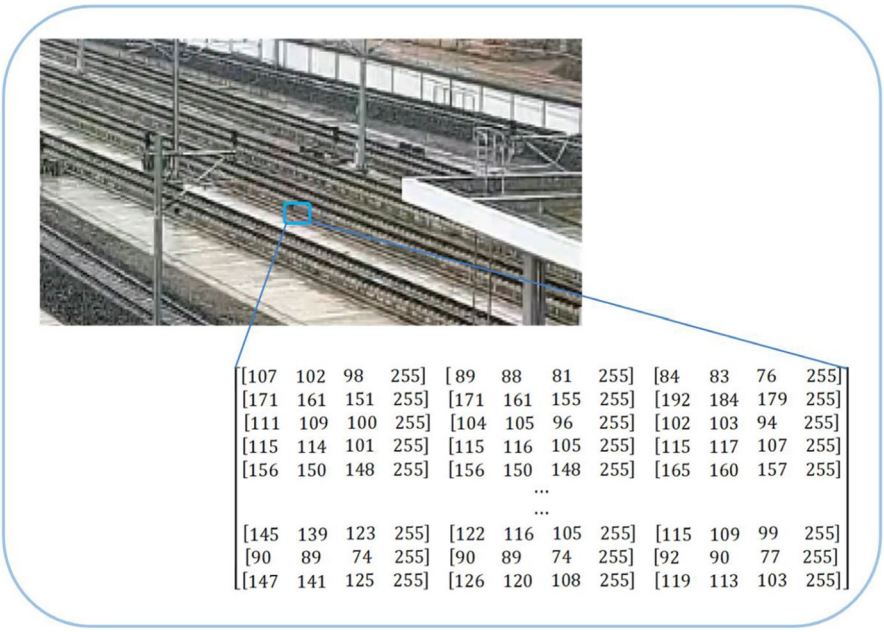

Fig. 3 Matrix storage form of digital images in a computer. Digital images are represented and stored in computer systems as matrices, providing a structured format for the storage and manipulation of visual data

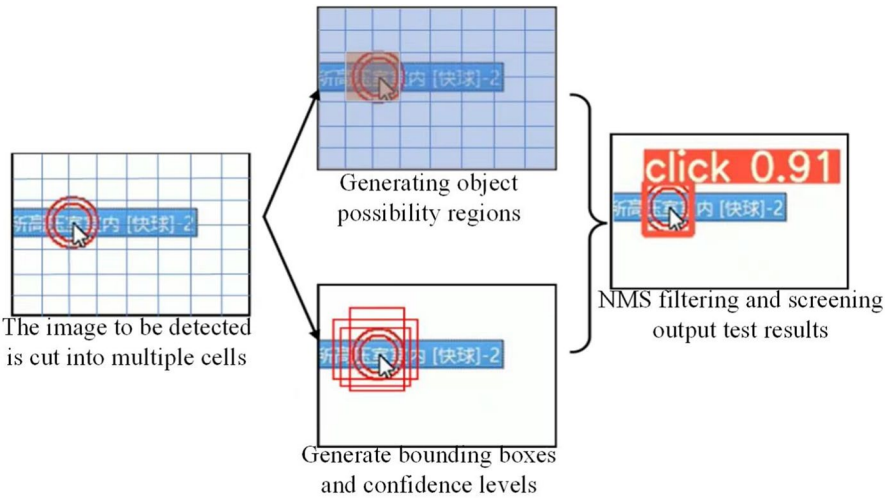

Fig. 4 YOLO-V5 model detection mouse click. The process uses YOLO-V5 to accurately measure the startup time of the video surveillance system and monitor the mouse click operations initiated by the tester. In addition, it employs grid-based object detection to efficiently and quickly identify objects

The video loading time measurement procedure consists of two main stages. Initially, frames from each image within the video are recognized. Before the user terminal initiates the video file, the video display area is typically black. Upon user initiation, the testing model continuously detects frames from the video display area on the user terminal, recording the display area images as RGB values within a matrix.

Subsequently, differences in average values of the R, G, and B channels between adjacent video image frames are compared, as delineated in algorithm 1 and corresponding to Fig. 2(c). To enhance the algorithm's efficiency, only the start time of the test (t_s) and the capture time of the current video display area (t_c) are retained.

To enhance the precision of the measurements, the algorithm assesses the difference between the average values of the R, G, and B channels between two successive image frames. This difference is then compared to a predefined threshold, with the comparison outcome stored in sequence M. If this difference is found to be greater than the threshold θ, the video is inferred to have been successfully captured. At this point, the first time (t_c) when the threshold is met is recorded as the end time of the measurement latency (t_e).

Through 500 threshold adjustment tests, it was found that the human eye can only perceive video frame changes when the absolute difference of the mean values of the three channels between consecutive video frames is greater than 30. Therefore, in this paper, the threshold is set as $\theta = 30$.

4.3 Feature matching-based measurement of response latency of PTZ camera

The paper utilizes feature matching based on ORB keypoints as an initial method to quantify the response latency of PTZ cameras. In PTZ camera control, the moment when the camera rotation is complete is considered the endpoint (t_e). To capture this moment, we depend on image detection technology. The area of image change on the PTZ video surveillance screen is monitored until it meets the satisfaction threshold φ. This is when we record the end time (t_e).

However, in the PTZ camera control response latency test, the camera's jitter may affect the data results at the end time (t_e). Therefore, we utilize image motion distance detection to eliminate the interference caused by camera jitter.

We utilize a feature matching method based on ORB keypoints to extract features from two frames of the image. First, we extract the feature parameters of the image and then calculate the spatial coordinate transformation parameters of the image. We set a threshold φ to exclude the interference of camera jitter, and the specific process for detecting the distance of image movement is illustrated in Fig. 5.

For image feature point matching, we use ORB feature points to characterize the image. ORB feature points are identified by detecting pixels in the image that significantly differs from surrounding pixels, which are considered keypoints. The BRIEF descriptor is then computed for each keypoint. We utilize the brute-force (BF) matching method for the initial feature matching. The method computes the distance

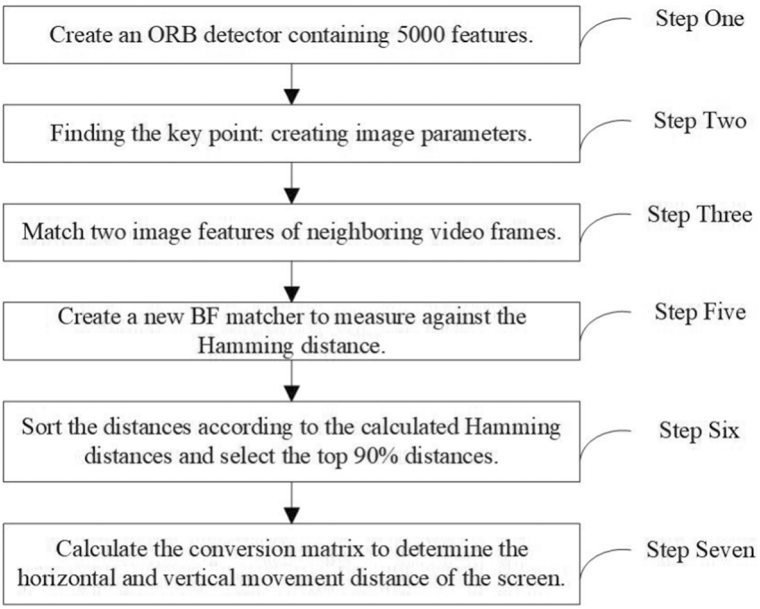

Fig. 5 PTZ camera movement distance detection steps based on feature matching. The process of determining the PTZ camera travel distance involves various steps focused on feature matching. The method eliminates the interference of camera jitter and measures the rotation response of the PTZ camera

between each descriptor in the training descriptor set and the query descriptor. The matching result is selected by ranking all the distances that meet the threshold φ requirement. By performing feature matching on video images, we can essentially eliminate the impact of camera jitter on the test results. This way, we record the first moment when the image on the video surveillance screen changes and meets the threshold $\varphi = 20$ as the end time (t_e).

4.4 Object detection-based response latency measurement for PTZ cameras

We will test the PTZ camera response by comparing the difference in contour area ratios of objects in the video frames before and after. This will be in addition to the methods that were introduced in 4.3. We will switch to the object contour area detection method when the number of feature point matches detected by the feature matching method falls below the threshold. This is because we need to consider the situation when a large occlusion suddenly appears in front of the camera, leading to no feature keypoint match between the front and rear video frames of the PTZ camera. At this point, we designate the moment when the number of feature point matches appearing in the front and rear video frames falls below the threshold as $'t'_p$.

The specific detection process is shown in algorithm 1. The specific detection process is shown in algorithm 2.

To detect moving objects, we utilize OpenCV to distinguish the dynamic foreground from the static background. We compare the current frame with the frame assumed to be a static background to determine if a moving object has appeared in the region. We utilize a background subtraction algorithm based on the OpenCV library to achieve the separation of foreground (moving objects) and background (changes caused by other factors) in the PTZ video monitoring screen. However, direct pixel comparison can lead to false detections due to significant interference from lighting, shadows, and other factors.

Before calculating the contour area, we convert the image to grayscale and apply Gaussian blurring and binarization to obtain a clear binary map, reducing background interference. Using the "findContours()" and "drawContours()" functions of the OpenCV library, we can draw the contours of the objects in the foreground image. Finally, the area (C) of the video changing contours is calculated using the "contourArea()" function. The PTZ camera response is considered successful if the ratio of the difference in contour area between two neighboring foreground frames exceeds 25%. The end time (t_e) is defined as the same as $'t'_p$. We determine the PTZ camera control response latency by calculating the time difference between the end time (t_e) and the start time (t_s) in the video surveillance system. In other words, $t = |t_e - t_s|$.

5 Results and discussion

This section presents the experimental testing performed to validate the proposed performance testing method of the railway video surveillance system based on object detection. The assessment centered around two main areas: the accuracy of the start time (t_s) and end time (t_e) records.

To ensure the accuracy of recording the start and end times, we compared our measurements with traditional manual stopwatch readings during the test evaluation. We maintained consistency in the test environment, hardware configuration, testers, and data analysts to ensure repeatability and accuracy. We synchronized data obtained from both response latency test methods during the experiment for subsequent analysis and evaluation.

Algorithm 2 Object Detection-Based Response Latency Measurement for PTZ Cameras

Input: Video frame list $frames$, threshold τ, and start time t_s.
Output: PTZ camera response latency t or "No response detected".

Initialize $foreground\_frames$ to .
Initialize $background\_frame$ to the first frame of $frames$.
Initialize $moving\_objects$ to .
Initialize $response\_success$ to False.
Initialize $contour\_area$ to 0.
while there are frames in $frames$ **do**
  if number of feature point matches $< \tau$ **then**
    if occlusion detected **then**
      Apply background subtraction algorithm on $background\_frame$ and the current frame to get $binary\_image$.
      Use $findContours()$ to detect moving object contours in $binary\_image$.
      for each contour in detected contours **do**
        if contour area > 0 **then**
          Add contour to $moving\_objects$.
          Update $contour\_area$ with the contour area.
        end if
      end for
      if $moving\_objects$ is not empty **then**
        Calculate the area ratio $ratio$.
        if $ratio > 25\%$ **then**
          Set $response\_success$ to True.
          if t_e is not set **then**
            Set t_e to the current frame time.
          end if
          return $t = |t_e - t_s|$.
        end if
      end if
    end if
  end if
  if all frames in $frames$ processed and $response\_success$ is False **then**
    return "No response detected".
  end if
end while

5.1 Introduction of dataset

In order to establish a reliable and valid test method, we use the Python programming language with Pycharm, a commonly used Python IDE, to implement our method. In our experiments, we used a dataset of 10,000 offline screen-recorded

videos obtained from different railroad sections, different weather conditions, and scenarios. For this experiment, we collected a dataset of 1000 screen-recorded videos of monitoring system responses on various railway sections under different weather conditions and scenarios. The video dataset has a resolution of 1850×1080 and a frame rate of 25 fps. We used this dataset to conduct simulation experiments to evaluate the accuracy and efficiency of our proposed methodology for performance testing.

We created an offline operational behavior image dataset to assess the practical performance of the underlying behavior on the client, given the limited availability of the original operational behavior dataset. The image dataset comprises 6000 images of testers' actions. We used the LabelImg tool to label these images, and the labeled image information was saved as a.txt document. To train, tune parameters, and validate the YOLO-V5 model, we divided the labeled images into a training dataset, a validation dataset, and a test dataset in an 8:1:1 ratio.

The offline operational behavior image dataset plays a key role in this study, which contains 6000 still images, each capturing a specific operational behavior performed by the tester on the PTZ camera control interface, such as mouse clicks. These actions involve controlling, adjusting, and operating the surveillance system, and are intended to evaluate the actual performance of client-side user actions in order to gain insight into how users interact with the surveillance system. Compared to the recorded video dataset, the offline operation behavior image dataset presents static characteristics and focuses on capturing specific user operation behaviors, while the recorded video dataset demonstrates the dynamic response of the surveillance system in different scenarios. By utilizing these two datasets together, the experiments are able to evaluate the proposed performance testing methodology more comprehensively, taking into account the close correlation between system operations and actual user behaviors, thus providing a more in-depth assessment of the effectiveness of the performance testing methodology.

5.2 Timing accuracy of start times-based YOLO-V5 model

To conduct the experiment, we configured the experimental parameters as specified in Table 1.

Table 1 Parameter setting of YOLO-V5 test model

Parameter	Setting
Input resolution	640×640
Number of classes	80
Model depth multiple	0.33
Layer channel multiple	0.5
Batch size	32
Number of epochs	300
Confidence threshold	0.4
IOU threshold for NMS	0.5

Table 2 Start time accuracy

Test items	Our method	Traditional method
Real-time video call	96%	95.4%
PTZ camera response	96.4%	95.8%

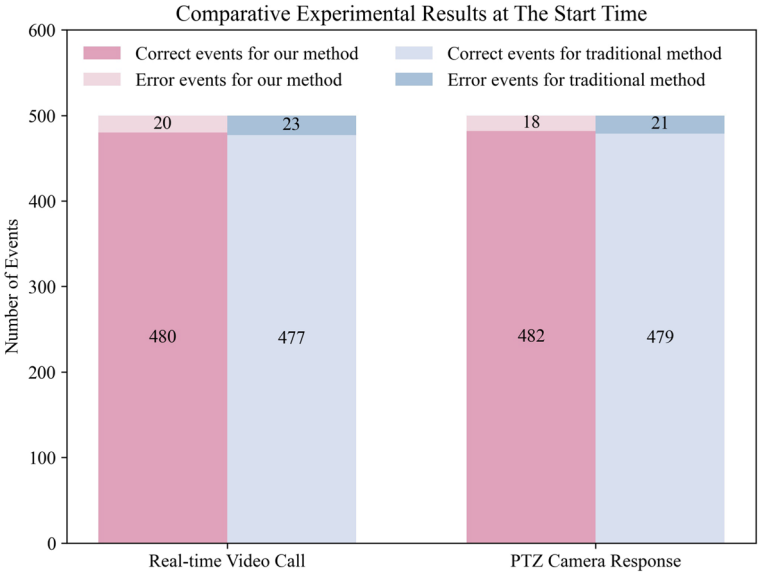

Fig. 6 Percentage of detailed data in different events at start time. We conducted a comparative experiment to assess the accuracy of the start moments. Comparing the results enables us to analyze the precision and effectiveness of the proposed testing method

The initial segment of our experimentation focused on evaluating the YOLO-V5 model's proficiency and precision in discerning user mouse maneuvers. A multitude of tests were orchestrated across a spectrum of conditions to emulate real-world environments. The test results are analyzed for two scenarios: real-time video call and PTZ camera response. Subsequently, we analyzed the data detected by the YOLO-V5 model at the moment of the operational command placement.

Comparison experiments were conducted 500 times with millisecond accuracy in timestamp recording. The test results are shown in Fig. 6, and the test accuracy is presented in Table 2. In the real-time video call scenario, the traditional manual reading method yielded 477 correct test data, while the method proposed in this paper produced 480 correct test data. For the PTZ camera control response scenario, the traditional manual reading method yielded 479 correct test data, whereas the proposed method yielded 482 correct test data.

From these empirical results, we obtained a measurable understanding of the YOLO-V5 model's operational accuracy and responsiveness.

Table 3 End time accuracy

Test items	Our method	Traditional method
Real-time video call	95.4%	95.6%
PTZ camera response	96%	96.4%

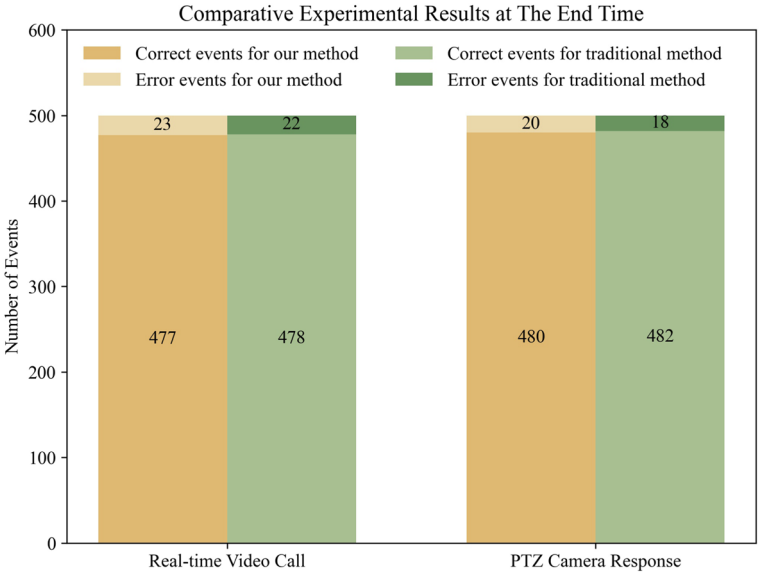

Fig. 7 Percentage of detailed data in different events at end time. We conducted comparative experiments to evaluate the accuracy of the final moments. The comparative results enable us to analyze the accuracy and validity of the proposed test method

5.3 Timing accuracy of end times based on object detection

The third part of the experiment assessed the accuracy of the object detection method at the moment of determining the end. This phase of the test is involved. The test results are analyzed below for two scenarios: analyzing real-time video call and PTZ camera response, respectively. We compared the measurements with traditional manual stopwatch readings during the test evaluation.

The comparison experiments were conducted 500 times with a recording accuracy of timestamps in milliseconds. The test results are shown in Table 3 and Fig. 7. In real-time video calling scenarios that utilize the sum of the differences of the mean values of the three channels of the image RGB to determine the end time (t_e), the traditional manual reading method yielded 478 correct test data, whereas the method proposed in this paper yielded 477 correct test data. For the PTZ camera control response scenario that uses a combination of feature matching and contour detection to determine the end time (t_e), the traditional manual reading method yielded 482 correct test data, whereas the method proposed in this paper yielded

480 correct test data. It should be noted that the average response latency of the PTZ camera control in these 500 tests is 189 ms, which meets the requirement of the relevant technical specification standard that mandates completion within 300 ms. In terms of accuracy, the method proposed in this paper is very similar to the traditional manual reading method.

In terms of overall testing time, the traditional manual reading method takes 10 h, while the method proposed in this paper takes only 4.5 h. Since the method in this paper fully utilizes computer vision for object detection, enhances testing efficiency, and reduces testing costs, it is more advantageous from a comprehensive perspective. The method proposed in this paper fully utilizes computer vision to enhance testing efficiency and reduce testing costs. Overall, this approach has multiple advantages and appears to be a more comprehensive solution.

5.4 Discussion on effectiveness and viability

Once the experiments were completed, the collected data were subjected to rigorous analysis to extract meaningful insights into the effectiveness of the proposed performance testing methodology, focusing on its precision and accuracy in comparison with conventional techniques. According to the test results, the proposed method demonstrates a high level of accuracy in measuring the response latency in the railway video surveillance system. It meets the test standard requirements and provides accurate results. In comparison with the traditional manual reading approach, using a latency-based approach to video surveillance systems based on object detection is not only more efficient but also more cost-effective.

Object detection-based response latency measurement method for video surveillance systems is an efficient solution to the overuse of generic testing tools in response latency testing by the railway video intermodulation comprehensive test team. The solution not only eliminates compatibility issues faced by testing tools tools, but also simplifies the process, making it more efficient. From a usability standpoint, testers conducting video surveillance system response latency measurements will find it easy to deploy this all-in-one tool on the video inspection system interface host device.

6 Conclusion

This paper introduces a novel approach for assessing the response latency of video surveillance systems through object detection. The control response latency of a video surveillance system is determined by calculating the time difference between the end time (t_e) and the start time (t_s). Our method pinpoints latency by tracking changes in video screen parameters post-mouse click, utilizing the YOLO-V5 model to identify the command initiation (t_s). To obtain accurate video response times (t_e), we leverage RGB channel analysis for the real-time video call's successful transmission moment. Feature matching and contour detection further refine the precision of latency measurements for PTZ camera controls.

Our experiments demonstrate that this automated method significantly outperforms traditional manual latency assessments in both accuracy and efficiency. This advancement is particularly beneficial for practical applications, streamlining the testing process for video surveillance systems and offering a reliable performance benchmark for integrated systems, such as those used in the railway industry. The efficiency and reliability gains are crucial for enhancing the overall performance and safety of these systems.

Looking ahead, future work will concentrate on refining the detection algorithms to minimize false alarms and omissions, and on extending the technique's applicability to diverse scenarios and systems.

Author's contribution Jia Miao conceptualized the study and contributed to the original manuscript; Li Zhu and Hongli Zhao contributed to the formal analysis of the study and participated in the revision of the original manuscript; and Sen Lin and Xinjun Gao contributed data as well as analyzed the methodology and validated the results.

Funding This work was supported Beijing Natural Science Foundation (L211002).

Availability of data There are no data available.

Declarations

Conflict of interest The authors have nothing to declare.

References

1. Liang H, Zhu L, Yu FR (2024) Collaborative edge intelligence service provision in blockchain empowered urban rail transit systems. IEEE Internet Things J 11(2):2211–2223
2. Li Y, Zhu L, Wang H, Yu FR, Tang T, Zhang D (2023) Joint security and resources allocation scheme design in edge intelligence enabled cbtcs: a two-level game theoretic approach. IEEE Trans Intell Transp Syst 24(12):13948–13961
3. Zhu L, Liang H, Wang H, Ning B, Tang T (2022) Joint security and train control design in blockchain-empowered cbtc system. IEEE Internet Things J 9(11):8119–8129
4. Liang H, Zhu L, Yu FR, Wang X (2023) A cross-layer defense method for blockchain empowered cbtc systems against data tampering attacks. IEEE Trans Intell Transp Syst 24(1):501–515
5. Zhu L, Li Y, Yu FR, Ning B, Tang T, Wang X (2021) Cross-layer defense methods for jamming-resistant cbtc systems. IEEE Trans Intell Transp Syst 22(11):7266–7278
6. Elharrouss O, Almaadeed N, Al-Maadeed S (2021) A review of video surveillance systems. J Vis Commun Image Represent 77:103116, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320321000729
7. Palatty JJ, Edireswarapu SPC, Sivraj P (2019) Performance analysis of freertos based video capture system. In: 3rd International Conference on Electronics. Communication and Aerospace Technology (ICECA), pp 595–599
8. Zhang Z, Lu W, Sun W, Min X, Wang T, Zhai G (2022) Surveillance video quality assessment based on quality related retraining. IEEE Int Conf Image Process (ICIP) 2022:4278–4282
9. Muller-Schneiders S, Jager T, Loos H, Niem W (2005) Performance evaluation of a real time video surveillance system. IEEE Int Workshop Vis Surveill Perform Eval Tracki Surveill 2005:137–143
10. Elharrouss O, Almaadeed N, Al-Maadeed S (2021) A review of video surveillance systems. J Vis Commun Image Represent 77:103116
11. Jansen J, Bulterman DCA (2013) User-centric video delay measurements. In: Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, ser.

NOSSDAV '13. New York, NY, USA: Association for Computing Machinery, p 37–42. [Online]. Available: https://doi.org/10.1145/2460782.2460789
12. Hill R, Madden C, Hengel Avd, Detmold H, Dick A (2009) Measuring latency for video surveillance systems. In: *2009 Digital Image Computing: Techniques and Applications*, pp 89–95
13. Wu Y, Bai X, Hu Y, Chen M (2022) A novel video transmission latency measurement method for intelligent cloud computing. Appl Sci, 12(24) [Online]. Available: https://www.mdpi.com/2076-3417/12/24/12884
14. Ultralytics (2020) "Yolo-v5," https://github.com/ultralytics/yolov5/releases/tag/V3.0. Accessed on 13 August 2020
15. Ubik S, Pospíšilík J (2021) Video camera latency analysis and measurement. IEEE Trans Circuits Syst Video Technol 31(1):140–147
16. Yi J, Luo S, Yan Z (2019) A measurement study of youtube 360 live video streaming. In: Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp 49–54
17. Siekkinen M, Kämäräinen T, Favario L, Masala E (2018) Can you see what i see? Quality-of-experience measurements of mobile live video broadcasting. ACM Trans Multimedia Comput Commun Appl (TOMM) 14(2s):1–23
18. Edwards D, Imming S (2023) Sports production through ai-powered sports action tracking and ptz cameras. SMPTE Motion Imag J 132(10):6–12
19. Technical Specification for Railway Integrated Video Surveillance System: Q/CR 575-2022, China National Railway Group Co. Std. (2022)
20. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 779–788
21. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: an efficient alternative to sift or surf. Int Conf Comput Vis 2011:2564–2571
22. Xie Y, Wang Q, Chang Y, Zhang X (2022) Fast target recognition based on improved orb feature. Appl Sci 12(2) [Online]. Available: https://www.mdpi.com/2076-3417/12/2/786
23. Yang Y, Wang X, Wu J, Chen H, Han Z (2015) An improved mean shift object tracking algorithm based on orb feature matching. In: The 27th Chinese Control and Decision Conference, CCDC). IEEE 2015:4996–4999
24. Dubrofsky E (2009) Homography estimation. Diplomová práce. Vancouver: Univerzita Britské Kolumbie, vol 5
25. Bai L, Zhao T, Xiu X (2022) Exploration of computer vision and image processing technology based on opencv. In: International Seminar on Computer Science and Engineering Technology (SCSET). IEEE 2022:145–147
26. Pandey H, Choudhary P, Singh A (2022) Detect and track the motion of any moving object using opencv. In: Cyber Security in Intelligent Computing and Communications. Springer, pp 355–363