IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 July 2023, accepted 18 July 2023, date of publication 21 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3297643

APPLIED RESEARCH

SCSS: An Intelligent Security System to Guard City Public Safe

KUN XIA[1], LINGXIANG ZHANG[1,2], SHUAI YUAN[1,2], AND YANG LOU[1]
[1]Department of Electrical Engineering, University of Shanghai for Science and Technology, Yangpu, Shanghai 200093, China
[2]National Local Joint Engineering Laboratory of High Energy Saving Motor and Control Technology, Anhui University, Hefei 230039, China

Corresponding author: Lingxiang Zhang (zlingxiang0122@163.com)

This work was supported in part by the National Local Joint Engineering Laboratory of High Energy Saving Motor and Control Technology Open Subject under Grant KFKT202105.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Experimental Committee of the University of Shanghai for Science and Technology.

ABSTRACT Traditional security surveillance detection relies on post-event forensics or is hosted on a backend server, making it impossible to identify behaviors filmed in the field online. This paper proposes the Smart City Security System (SCSS) for detecting anomalous activity in public locations online. SCSS combines the DeepSORT and YOLOv4 algorithms to generate the DS-YOLO aberrant behavior detection algorithm, which compares and matches the target detected in the previous picture frame with the target detected in the following frame to achieve detection and tracking. SCSS is equipped with GPS, WIFI, and Uninterruptible Power Supply (UPS). When a risky behavior is detected, the system will upload the abnormal event as well as the latitude and longitude that occurred to the cloud via the WIFI and notify the user. The recognition accuracy of three deviant behaviors, including Fight, Car Accident, and Fall, was examined using diverse situations, and the results were 89%, 90%, and 90.33% respectively. The findings demonstrate that SCSS has successfully made the transition from passive monitoring to active identification, offsetting the flaws of conventional security systems that can only post-mordem forensics, and bridging the gap of the construction of national smart cities.

INDEX TERMS Intelligent security systems, image recognition, deep learning, DS-YOLO.

I. INTRODUCTION

In recent years, the issue of urban public safety has received increasing attention. With the ''safe city'', ''smart city'' and other urban construction projects proposed, the establishment of a sound smart city security system also came into being.

Current urban security monitoring methods focus on the performance of the front-end camera, staying in the stage of information acquisition, passive monitoring, often committed to the capture of video transmission to the cloud before the next step analysis, but limited exploration of vertical areas, and not in the field of timely online analysis of the video [1], [2], which lead to increased communication overhead, and can not quickly make security measures. In addition, due to the complex environments and dense crowds in urban public places, the identification of abnormal behavior accuracy rate is also undesirable.

This study proposes Smart City Security System (SCSS) based on the Jetson Nana demo board working with several interactive sub-modules, which can identify abnormal behaviors to be detected online (Fight, Car Accident and Fall), transfer to the cloud and synchronize to the client. The SCSS integrates data transmission, video, alarm and control, turning The SCSS integrates data transmission, video, alarm and control, turning the traditional security system from post-mordem forensics to real-time tracking, analysis, stopping and alarming, effectively reducing the incidence of dangerous events in the city and reducing the loss of people's life and property.

SCSS uses YOLOv4 as a framework for detection and captures video in real time through cameras. To address the issues of target obscuration and information loss caused

The associate editor coordinating the review of this manuscript and approving it for publication was Sangsoon Lim.

by complex environments in public places, the DeepSORT algorithm is combined with YOLOv4 to form DS-YOLO abnormal behavior detection algorithm, which analyses the captured footage frame by frame and compares and matches the target detected on one frame with the next, thus providing real-time detection and tracking of the target [3]. In order to respond quickly to the detected abnormal behavior, SCSS has built-in GPS and WIFI module. Via the MQTT to transmit the latitude and longitude information obtained by the GPS module in real time, helping the user side to obtain the location of the abnormal behavior in the first time and to arrive at the scene in time to deal with it. In that case, SCSS ensures timely security work and prevents the situation of no on-site personnel to help with the alarm when dangerous behavior occurs in remote streets.

The contributions of this paper are as follows:

This paper presents SCSS, which detects abnormal behavior such as violence in public places, car accidents and falls in real time.

Online recognition is performed via Jetson Nano, eliminating the need to transfer to the backend for further batch detection.

The DeepSORT algorithm is added to the YOLOv4 framework, and the TensorRT optimizer is used to improve the efficiency and accuracy of abnormal behavior detection.

SCSS is embedded with GPS and utilizes IoT technology to connect multiple platforms, enabling information interfacing between embedded devices, mobile phones and cloud databases.

Through multiple application scenarios, SCSS has proven to be effective in providing intelligent security for cities through four main steps: online target detection, target tracking, abnormal behavior detection and information transmission, contributing to the construction of ''safe cities'' and ''smart cities''.

The rest of the paper is organized in the following way. Section II introduces the latest research advances in smart security systems and anomalous behavior detection in cities. Section III introduces the system. Section IV introduces the DS-YOLO combination algorithm involved in this system. Section V verifies the performance of the proposed system through experiments in real scenarios. Section VI summarizes the current research progress and provides an outlook for the future.

II. RELATED RESEARCH

Intelligent monitoring of abnormal behaviors is a means of achieving smart security in cities [4], [5], [6]. In this section, we present the current state of research on video surveillance for abnormal behaviors in three areas: Car accident, Fall, and Fight.

Wang et al. [7] proposed an improved region proposal network based on the Faster Region based Convolutional Neural Network (Faster RCNN). The network employs four strategies to improve the Region Proposal Network (RPN), and the network structure is Multi-strategy Region Proposal Network (MSRPN). Experimental results show that the MSRPN algorithm has good performance in small target detection and is faster than other target detection algorithms, but it is less effective for some large vehicles detection. Zhang et al. [8] proposed a vehicle impairment detection segmentation algorithm based on migration learning and an improved Mask RCNN (Mask Region Convolutional Neural Network). It is trained by self-made dedicated dataset. Test results show that the improved Mask RCNN has better average precision (AP) values, detection accuracy and masking accuracy, but the mask instance segmentation cannot be completely correct, and some areas in which the damage is not obvious cannot be segmented. Fang et al. [9] proposed a self-supervised consistency learning framework (SSC-TAD). Traffic accidents are detected by considering temporal frame consistency, temporal object location consistency and spatial-temporal relationship consistency of road participants. The effectiveness of the scheme is verified through an exhaustive evaluation of two large datasets, namely the AnAn Accident Detection (A3D) dataset and the recently collected DADA-2000 dataset, but the AP is underperforming. Tian et al. [10] proposed an automatic car accident detection method based on CVIS and built a novel image dataset CAD-CVIS. they utilized Multi-Scale Feature Fusion (MSFF) and loss function with dynamic weights to enhance the performance of detecting small objects. However, images need to be captured by roadside cameras and transmitted to the server for incident detection, which requires a certain amount of communication costs. He et al. [11] used a constrained least squares algorithm to remove motion blur and applied Kalan filtering to the sharpening process to eliminate noise blur. The results of the study present that such intelligent video surveillance techniques can effectively improve the level of intelligent video surveillance technology, reducing the analysis time by half and greatly reducing the incidence of traffic accidents, but constant adjustment and verification of the surveillance situation is required to establish an intelligent surveillance system.

Lin et al. [12] proposed a fall detection system with neuromorphic computing hardware for AI-based edge computing. The images of individuals were captured through the camera and transmitted to the neural network model on the edge computing platform. After the detection of the object characteristics, the SVM was used for classification. However, fall detection is not effective in the presence of obstructing persons. Kim et al. [13] collected data on human falls occurring in four directions while walking or standing, and developed a center of mass (COM) based fall recognition system. It was experimentally confirmed that the recognition rate of both the convolutional neural network learning model and the long and short-term memory learning model exceeded 94% on the embedded platform (Jetson TX2). However, this study required the collection of a large amount of data on falls from different directions, which are not easily collected. Lu et al. [14] designed an Image-Based Fall Detection System (IFADS) for nursing homes, which focuses on falls that occur

when the old sitting down and getting up from a chair. IFADS first applies an object detection algorithm to identify people in video frames. A posture recognition method is then used to keep track of the person by checking the relative position of the chair and the person. Hence, an alarm is triggered when a fall is detected. Experimental results show that IFADS achieves an average accuracy of 95.96%, but the application scenario for IFADS is only limited to nursing homes and is not applicable in public place scenarios. Yu et al. [15] proposed a fall detection method based on joint learning and extreme learning machines. The experimental results show that the Fed-ELM has a 6.16% increase in accuracy compared to the direct use of the trained ELM. However, the data used was collected in an experimental environment and could not include all real-world data. The proposed algorithm requires the user to manually mark the incorrect data, which can easily result in incorrect markings and thus lead to a reduction in the quality of Fed-ELM.

Sun et al. [16] put forward a Multi-View Maximum Entropy Discrimination (MVMED+) model that combines various features of an image and thus uses complementary information between views to classify the image. For efficient optimization, Sun further derives a sequential minimum optimization algorithm to train the model. And the model was experimentally verified to outperform both the traditional single-view approach and the more recent multi-view approach, but the performance of the model can be affected if there is no strong consistency between different views. Ullah et al. [17] proposed a computationally intelligent Violence Detection (VD) method. First, the video stream acquired through a visual sensor is processed by a lightweight Convolutional Neural Network (CNN) and then temporal optical flow features are extracted from the information shots. Finally, a final feature map is generated by inserting a multi-layer long and short-term memory network to learn the violence patterns in the frame sequences. The results validate that there is some improvement in the accuracy of the method. However, the accuracy obtained in outdoor monitoring is still only 49%. Ye et al. [18] used image features and acoustic features for campus violence detection and proposed an improved Dempster-Shafer (D-S) algorithm. Using C3D (Convolutional 3D) neural network for feature extraction and classification, the recognition accuracy of the improved algorithm was improved by 10.79%, with a final recognition accuracy of 97.00%. However, the application scenario was limited to campus.

The functions proposed in this paper are based on image recognition to identify anomalies in Car accident, Fall, and Fight, and combination of YOLO and DeepSORT is used to improve the accuracy of the recognition in order to solve the problem of occlusion in crowded environments [19], [20], [21].

III. PROPOSED FRAMEWORK

SCSS serves two main functions: abnormal behavior detection and real-time monitoring with alarms. This chapter introduces the composition and design of the system from three aspects: hardware design, software design, and user interface design. The system is illustrated in Fig. 1, which shows how SCSS monitors video through cameras, analyzes video screens, and uploads information about abnormal events and their locations to the cloud and the user when detected.

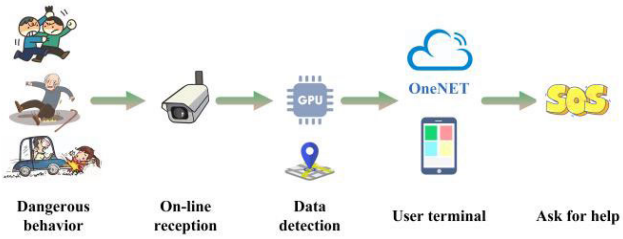


FIGURE 1. Overall system block diagram.

A. HARDWARE OVERVIEW

SCSS is designed with a modular architecture and is composed of Jetson Nano, video capture module, WIFI module, GPS module, and UPS. The hardware components of SCSS are illustrated in Fig. 2

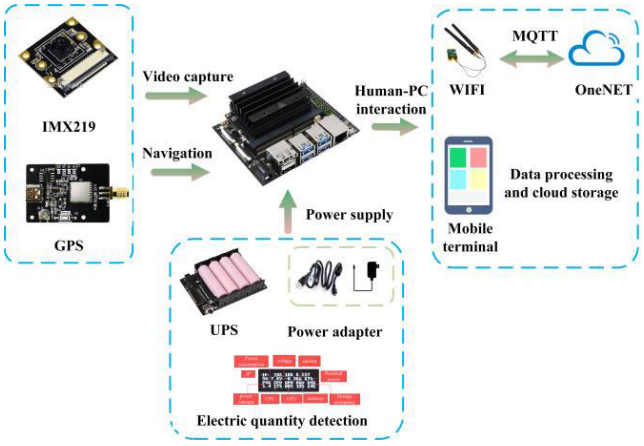


FIGURE 2. The hardware composition of SCSS.

The Jetson Nano serves as the control center of the entire system, responsible for scheduling platform resources and running deep learning models to detect multiple targets in complex environments. It also transmits hazard information to the cloud. Besides, it deducts and recognizes of the collected dates through camera, and finally transmitting the target information to the OneNET. The GPS module locates the latitude and longitude of the incident, while the WIFI communication module enables cloud connectivity. If the Jetson Nano identifies dangerous behavior, it can transmit the corresponding information and incident location accurately and simultaneously to the OneNET cloud. Additionally, the power management module includes UPS and power detection unit that communicates via the I2C interface. This

unit measures battery voltage, current, power, and remaining power to ensure the stable operation of the intelligent security system.

The Jetson Nano development board is an affordable AI development board designed by NVIDIA. It features a quad-core Cortex-A57 processor chip, 4 GB LPDDR memory, and 128-core Maxwell GPU. It can run multiple algorithms and AI frameworks, such as TensorFlow, Keras, PyTorch, Caffe, etc. It supports NVIDIA JetPack and supports multiple neural networks running in parallel to achieve image classification, face recognition, speech processing, target detection, and object recognition tracking, etc. It is suitable for developing small structure, low-cost, and low-power consumption devices. The Jetson Nano has 5-10 W power and 473 GFLOPS total computility and comes with a TensorRT optimizer that allows training model files to be placed directly into TensorRT without relying on deep learning framework.

The camera uses SONY IMX219 and has a wide field of view of up to 160°. It measures just 25 mm × 4 mm × 20 mm, features 800 W pixels, a resolution of 3280 × 2464, and can operate normally at temperatures ranging from -20 ° to 60 °.

The WIFI module utilizes the 8265NGW with M.2 interface and supports 2.4 GHz/5 GHz dual-band WIFI and Bluetooth 4.2. This module enables easy networking via WIFI, which facilitates the transmission of target information from the Jetson Nano to OneNET.

The GPS module is powered by ATGM336-5N, which is compatible with 4G/3G/2G communication and GNSS positioning, making it suitable for global positioning and other functions when integrated with the Jetson Nano. It supports GPS, BeiDou, Glonass, and QZSS base station positioning, and can be easily connected to the Jetson Nano PWR or 5 V port via a jumper for automatic power-on.

The power module of the intelligent security system can be powered by the 5 V/4 A power adapter and is equipped with UPS, which is located directly under the motherboard. The UPS allows for simultaneous charging and discharging, ensuring uninterrupted power supply to the Jetson Nano. A 5 V voltage regulator chip provides sufficient power to the Jetson Nano with stable output of 5 V and maximum current of 5 A. The battery voltage, current, power, and remaining charge are measured via the I2C interface, and the program outputs parameters such as the percentage of battery power remaining while running. The program is designed to detect low battery levels and control the Jetson Nano to save data before shutting down, preventing sudden power failure and data loss.

The photo of the hardware composition is shown in Fig. 3.

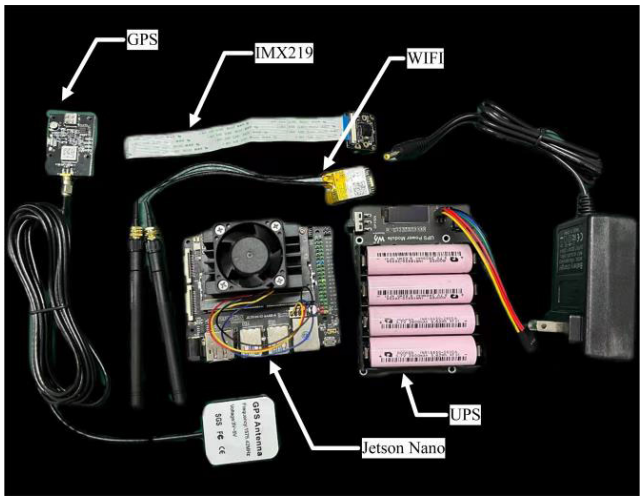


FIGURE 3. Photo of hardware composition.

B. SOFTWARE WORKFLOW

The software flowchart, as illustrated in Figure 4, adopts modular design approach to minimize the coupling between various systems. The program comprises several subroutines, including the power detection subroutine, the WIFI module subroutine, the GPS positioning subroutine, the camera acquisition subroutine, and the image recognition subroutine. The main program invokes each of these subroutines based on the system logic.

The SCSS is capable of realizing one-key boot function, with the option of powering the system via power adapter or UPS. It also includes the ability to detect power and power consumption parameters and initialize the monitoring unit and communication unit.

After the SCSS is started, the video surveillance module starts to capture the abnormal behavior in the surveillance screen and returns the recognition result to Jetson Nano. after the SCSS completes the network connection via WIFI, it can connect to the designated OneNET platform, and the device can send and receive data with the OneNET platform after the connection is completed. At the same time, the GPS module will complete the GPS data collection and upload to the platform through the 8265NGW module with the detailed location and latitude and longitude of the abnormal event.

The OneNET platform can store and analyze the collected data and issue alarm notifications to the client side, and the client can remotely monitor the situation through the WeChat, which supports real-time video surveillance and playback, as well as the viewing of historical data and event records. The client can also control the system to take corresponding measures when necessary, such as sending warning message, sounding an alarm, or taking other actions to prevent or mitigate the abnormal situation [22].

C. USER-SIDE DESIGN

The connection between the device data and the cloud platform and client is established through the use of data streams, which is a device property that serves as the target object for the platform's services, including rules, triggers, and message queues. It is used to categorize device data. The cloud platform receives data in a key-value format, where the value can be in a range of user-defined formats, such as floating-point, integer, string, and json. Data points are stored in the

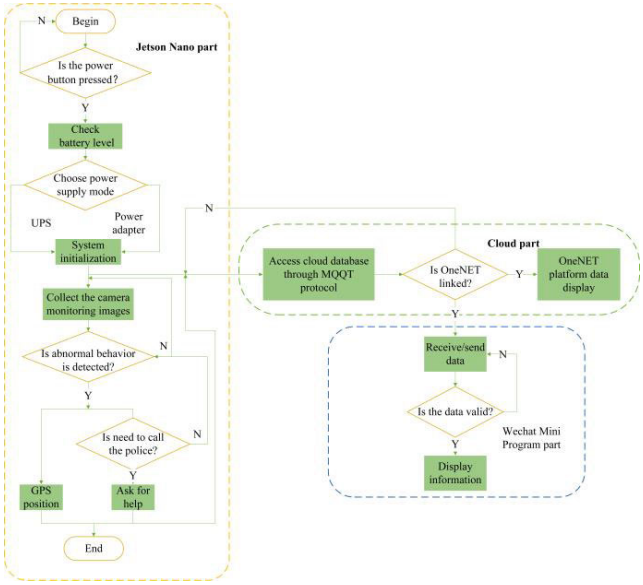


FIGURE 4. SCSS software flow chart.

data stream in temporal order, allowing for easy access and analysis of historical data.

The data stream with sensors is responsible for storing data collected by sensors, such as latitude and longitude collected by the GPS module and image data collected by the camera module. The stream with abnormal alarms is responsible for storing the alarms of identified abnormal events. The data in the stream is finally displayed by the visual application control, in which the GPS stream is called as ''location'' and the camera stream is nominated as ''image''. Besides, the names of the abnormal alarm streams are shown in Table 1.

TABLE 1. OneNET platform abnormal alarm data flow name.

Data stream name	Alarm conditions	Description
Normal	0	No abnormal behavior detected
Abnormal	1	Have a fighting accident
Abnormal	2	Have a fall accident
Abnormal	3	Have a car accident

The data points are stored in the data stream and can be displayed in the form of a list, with functions such as data history query, Excel data export, and real-time data refresh. This enables real-time monitoring of the current environment and provides data support for big data analysis of the security situation. The exported images can also be used as data for abnormal event recognition.

The WeChat applet is provided on the cell phone to enable users to monitor their target devices anytime and anywhere, regardless of geographic location, and to facilitate the detection of anomalies and alerts [23]. When designing the WeChat applet, after creating the graphics and layout of the display interface with HBuilder X, the users can edit the properties and styles of the corresponding controls, associate the corresponding data streams, select alerts on WeChat for detected dangerous behaviors, and view historical report data. The WeChat applet interface is shown in Figure 5, which consists of the login Page, and the home page. From the login page, you can click on ''Monitor'' to go to the login page and ''History'' to view the monitoring history. On the login page, you can see the coordinates of the abnormal event and the scene, and you can call for help by clicking on ''SOS''.

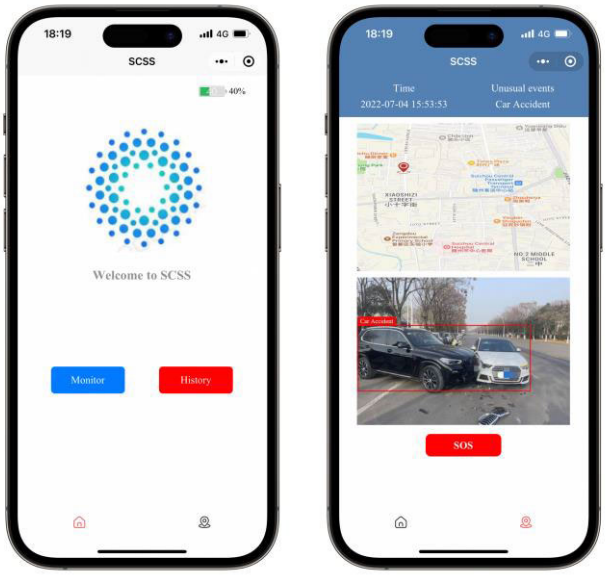


FIGURE 5. WeChat Mini Program interface.

IV. ABNORMAL BEHAVIOR RECOGNITION BASED ON DS-YOLO NETWORK MODEL

Due to the large flow of people in public places and the presence of obstacle occlusion, it is prone to trigger missed detection issues. In this paper, we design a target tracking model with the YOLOv4 algorithm as the detector to extract feature information. The network structure of DeepSort algorithm replaces the intersection ratio IOU matching by the generalized intersection ratio GIOU association matching to improve the matching between the target detection frame and the prediction frame, thus reducing the missed detection phenomenon.

A. DS-YOLO NETWORK MODEL

1) YOLOV4 NETWORK

The structure of the YOLOv4 model is shown in Figure 6. Compared to the YOLOv3 model with improvements on the backbone feature extraction network CSPDarknet-3 and the inclusion of the feature pyramid network SPPnet and PANet, the YOLOv4 model can be defined as YOLOv3 enhanced network with the backbone feature extraction network CSPDarknet-3 [24], in which the activation function of DarknetConv2D is modified from LeakyReLu to Mish function.

The Mish function is calculated in (1). The Mish function is overall smoother than the ReLU function, it is not overall

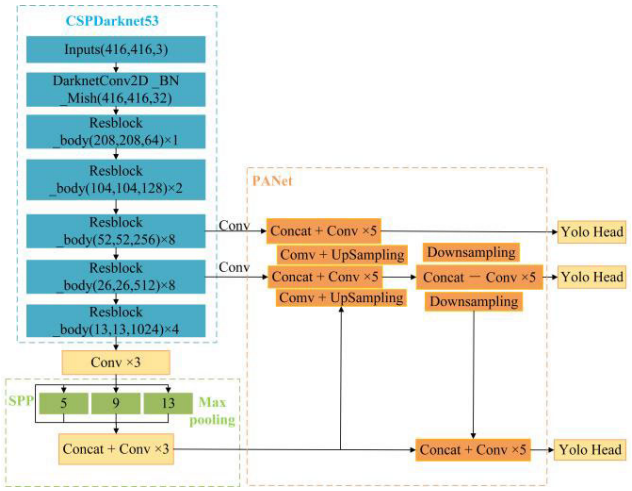


FIGURE 6. YOLOv4 network model.

forced zero bound like the ReLU function, and it allows some negative values when used as an activation function. For the activation function, the smoothing of the function can feed the data into the deep network better and improve the accuracy for the network model.

$$\text{Mish} = x \times \tanh\left(\ln(1 + e^x)\right) \tag{1}$$

where x represents the input tensor.

The structure of CSPnet was used in the network, which is shown in Figure 7. With the inclusion of the CSP structure, the feature mapping of the base layer is first divided into two parts [25], and then they are combined on a certain channel through a cross-stage hierarchy, which not only reduces the amount of computation but also ensures the accuracy of the model.

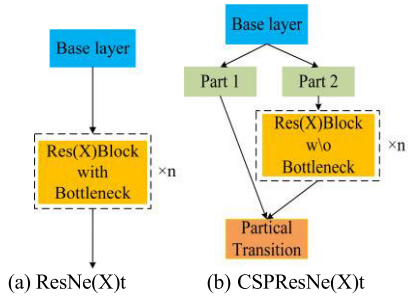


FIGURE 7. ResNet structure before and after the addition of CSP.

The enhanced feature extraction network in Yolov4 contains SPP and PANet networks, where the SPP structure is involved in the convolution of the last feature layer of CSPDarnet53. After three Leaky convolutions of the last feature layer of CSPDarknet53, respectively, using four different scales (13×13, 9×9, 5×5, 1×1) of the max-pooling is processed to be able to enhance the perception of the feature map and sort out the most significant features of the target in the feature map. PANet is essentially an instance segmentation algorithm, and its role in the network model is the iterative extraction of features, and feature fusion is performed in the structure for the effective feature layer of the backbone feature extraction network and the output feature layer of the SPP network [26]. In the structure, up-sampling, convolution and stacking operations are first performed on the effective feature layer and the output feature layer of the SPP, and then down-sampling, convolution and stacking operations are performed on the above operations with the aim of enhancing feature fusion and extracting more effective features.

The CIOU formula is shown in (2).

$$CIOU = IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{2}$$

where IOU is the intersection ratio between the prediction frame and the label frame, $\rho^2(b, b^{\text{gt}})$ represents the Euclidean distance between the label frame and the prediction frame, b and b^{gt} are the prediction frame and the label frame, and c represents the diagonal distance of the smallest closed area that can contain both the prediction frame and the label frame. Besides, αv is the aspect ratio influence factor, where α is used to balance the ratio and v is used to evaluate the consistency of the aspect ratio between the target frames.

The equations for α and v are shown in (3) and (4).

$$\alpha = \frac{v}{1 - IOU + v} \tag{3}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{4}$$

where w_{gt} and h_{gt} are the width and height of the label box, w and h are the width and height of the prediction box. The final 1-CIOU gives the corresponding LOSS. The final output data composition of the YOLOv4 model is the same as that of YOLOv3, and again the final output scales are 13×13, 26×26 and 52×52 feature maps respectively.

2) DEEPSORT ALGORITHM

SCSS uses the DeepSORT algorithm to track targets. It is found that the direct use of the SORT algorithm is more accurate and precise, but it is prone to target label switching, and is only more accurate when the target state is more deterministic while less effective when the target state uncertainty is high due to multiple targets overlap or occlusion. The DeepSORT algorithm replaces the association metric in the SORT algorithm based on a more stable metric, which increases the robustness of the algorithm model to missing and obstructed targets, and to a certain extent reduces the problem of target label switching in the algorithm task.

The DeepSORT algorithm continues the 8-dimensional state space $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ from the SORT algorithm, where (u, v) denotes the center coordinates of the target frame in the image, γ is the aspect ratio of the target frame, h is the height of the target frame, and $(\dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ corresponds to the velocity on the image coordinate system, using the above 8-dimensional state as a direct observation of the target state. For each trajectory k, the number of frames matched is

counted from the moment of the first match a_k, and the count is increased when a match is made during the Kalman filter prediction and reset to zero if the trajectory is associated with a new prediction [27]. At the same time, a life-cycle threshold A_{max} is also set, beyond which time no match is considered to have left the scene when the tracked object leaves the scene and is removed from the trajectory.

Since each newly detected object may become a new trajectory, if they are directly classified as a trajectory, wrong and false detections will occur frequently. DeepSORT marks the new detection as undetermined, then observes the next several frames, and if the next three consecutive frames are successfully matched, it is marked as determined and confirmed as a new trajectory. Otherwise, it is marked as deleted and is no longer considered to constitute a trajectory. The diagram of the state transition is shown in Figure 8.

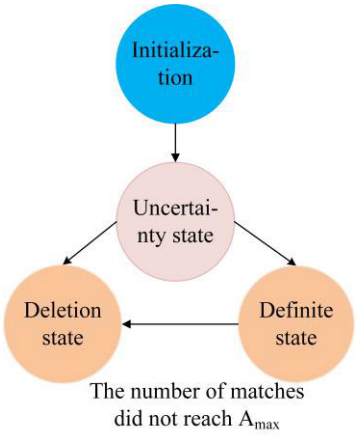


FIGURE 8. The diagram of the state transition.

In the DeepSORT algorithm, in addition to inheriting SORT's approach of using the Hungarian algorithm to match Kalman-predicted states with new detection results, information on appearance and motion is also integrated into the matching strategy, proposing Mahalanobis Distance, as shown in (5).

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right) \tag{5}$$

where $d^{(1)}(i, j)$ denotes Mahalanobis Distance, d_j denotes the state vector of the jth detection box (u, v, γ, h) and y_i denotes the predicted position of the ith tracker target and S_i is the covariance matrix of the observation space at the current moment obtained from the predictions of the trajectory Kalman filter.

The above equation indicates how well the jth prediction matches the ith trajectory. This matching takes into account the uncertainty in the state estimate. The 95% quantile of the cardinality distribution can be used as a threshold to exclude unlikely correlations. If the association between the ith trajectory and the jth prediction is acceptable, it is calculated as 1, as shown in (6).

$$b_{i,j}^{(1)} = 1[d^{(1)}(i,j) \le t^{(1)}] \tag{6}$$

where $b_{i,j}^{(1)}$ is an indicator that characterizes the relationship between the martingale distance and the threshold, and when it is less than the threshold, the match is successful, and $t^{(1)}$ is the threshold.

For motion target state with low uncertainty, the Mahalanobis Distance matching process described above applies to this approach, but motion target state estimation using the Kalman filter algorithm in the space where the frame image is located is only a compendiary prediction process. To address these issues, the algorithm introduces a second association determination condition that uses cosine distances to measure differences. For each detection frame d_j, the appearance descriptor r_j is computed, where $||r_j|| = 1$, and in addition, for each trajectory k, a gallery is used to store the nearest 100 appearance descriptors. Finally, the minimum cosine distance between the ith trajectory and the jth detection was measured, as shown in (7).

$$d^{(2)}(i,j) = min\left\{1 - r_j^T r_k^{(i)} | r_k^{(i)} \in R_i\right\} \tag{7}$$

where $d^{(2)}(i, j)$ denotes Cosine distance, r_j denotes the feature vector corresponding to the detection frame, $r_k^{(i)}$ is the set of eigenvectors of tracker corresponding to the nearest set of 100 frame eigenvectors, R_i denotes a vector library of appearance features.

Meanwhile, a limit is set for the metric. For example, if the association between the ith trajectory and the jth prediction is acceptable, then it is calculated as 1. The threshold here is different from the Mahalanobis distance, and the appropriate threshold needs to be found on a separate training dataset as shown in (8).

$$b_{i,j}^{(2)} = 1[d^{(2)}(i,j) \le t^{(2)}] \tag{8}$$

where $b_{i,j}^{(2)}$ is an indicator that characterizes the relationship between the Cosine distance and the threshold, and $t^{(2)}$ is the threshold.

The two different measures used above are linearly weighted as the final measure, as shown in (9), and the proportion of the two can be controlled by controlling the weighting factor.

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{9}$$

where $c_{i,j}$ denotes the linearly weighted metric, λ is the weighting factor.

The accuracy of the Kalman filter prediction is reduced when there is prolonged occlusion or overlap in the video [28]. If two tracking results are then competitively associated with the same detection result, the trajectory of the long-obscured or overlapping target in the image will be more inaccurate and the covariance distance will increase. The detection result completes its association with the trajectory of the long-overlapping or obscured target as a result of the trajectory of the long-obscured or overlapping target in the image not having received localization information for a long time, which then affects the issue of how long the target is tracked [29]. Cascade matching has therefore been added to

the algorithm to improve this situation, prioritizing the more common objects. The cascade algorithm flow is shown in Figure 9. The set of tracking frames for the deterministic state is T= {1, …, N}, the set of currently detected results is D= {1, …, M}, and the maximum mismatch value is A_{max}.

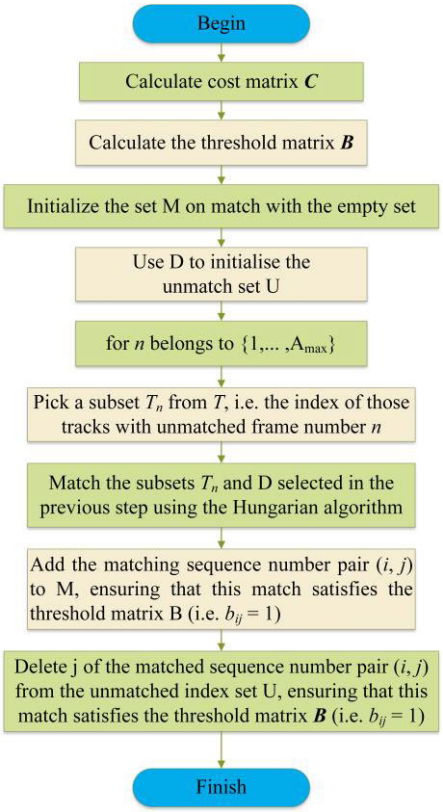


FIGURE 9. Cascade algorithm flow.

Firstly, the trajectory T, the detection D, and the maximum mismatch value A_{max} are entered into the cascade matching, the cost and threshold matrices are then calculated, the set of matches is initialized as M and the set of unmatched as U, the selected subsets T_n and D are matched by the Hungarian algorithm, add the matching sequence numbers (i, j) to M and ensure that the threshold matrix $\boldsymbol{B}$ is satisfied. Finally, delete the matching sequence numbers from the unmatched set U and ensure that this match satisfies matrix $\boldsymbol{B}$.

The appearance descriptor r_j uses a convolutional neural network trained on a large-scale dataset to extract the feature information of the target [30]. The structure of a residual network with two convolutional layers and six residual blocks was used in the algorithm to obtain features of dimension 128 via a fully connected layer. Finally, the 128-dimensional appearance features were extracted by training a deep network on the dataset, using L2 normalization to project the features onto the unit hypersphere so that they are compatible with the cosine metric, as shown in Table 2 for the specific network structure.

The overall flow chart of the DeepSORT algorithm is shown in Figure 10.

TABLE 2. The network model structure of appearance descriptors.

Network layer	Convolution kernel size/step size	Output scale
Conv1	3×3/1	32×128×64
Conv2	3×3/1	32×128×64
Max Pool 3	3×3/2	32×64×32
Residual 4	3×3/1	32×64×32
Residual 5	3×3/1	32×64×32
Residual 6	3×3/2	64×32×16
Residual 7	3×3/1	64×32×16
Residual 8	3×3/2	128×16×8
Residual 9	3×3/1	128×16×8
Dense10		128
Batch and L2 normalization		128

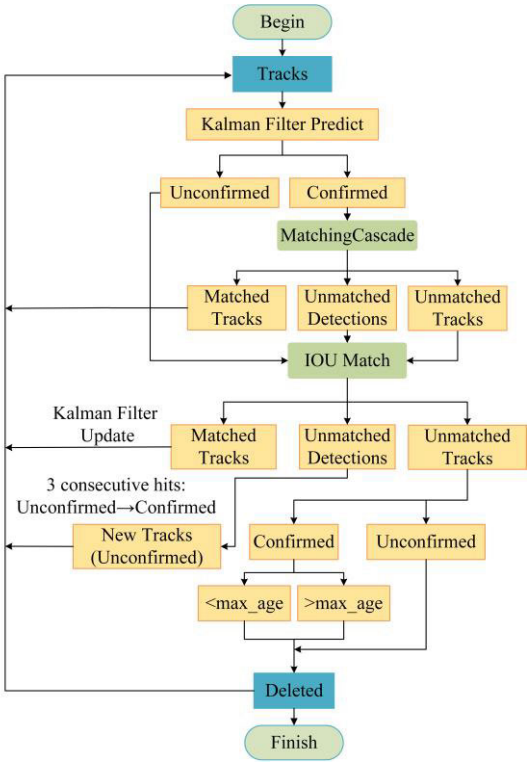


FIGURE 10. DeepSORT algorithm flow chart.

Firstly, the Kalman filter is used to predict the Tracks from the previous round to produce Confirmed Tracks and unconfirmed Tracks. The Confirmed Tracks are matched in cascade to produce three state results, the Unmatched Detections are matched again by IOU Match. While the unmatched Tracks that are not yet confirmed and those that are confirmed but exceed the threshold are deleted. Finally, the matched Tracks are merged and the Kalman filter is updated.

B. IMPLEMENTATION OF ABNORMAL BEHAVIOR RECOGNITION

Three types of abnormal behavior were selected and compared by Normal, namely Fight, Car Accident, and Fall. And more than three thousand annotated behavior recognition images were collected by camera in various scenes and

angles, each containing 500 to 700 images. The final model divides these sample data into training set, validation set and test set according to a ratio of 8:1:1. The specific behavioral sample data included in the abnormal behavior data samples are illustrated in Table 3.

TABLE 3. Abnormal behavior sample.

Behavior type	Number of training samples	Verification sample	Test sample
Normal	581	96	96
Fight	595	98	98
Car Accident	564	94	94
Fall	594	99	99

The images were named and organized into folders and labelled using labellmg software, and the dataset is shown in Figure 11. Besides, the Pytorch machine learning framework was built and the GPU environment was configured, the GPU used was GTX3060.

FIGURE 11. The collected dataset under different scene conditions.

For the DS-YOLO algorithm some parameters need to be set, setting the iteration ordinal number to 120. The tracker status changes from unconfirmed to confirmed only when the target is matched more than three times in a row, such as n_init=3. The specific configuration of the model training hyperparameters is shown in Table 4.

In order to reduce the training time of the model and accelerate the iterative convergence of the network, the model was set at epoch 120. Figure 12 presents the change of the loss function of the DS-YOLO network during the whole training process. It can be directly seen in the figure that the training loss value of the model decreases during the training process from the 1st to the 10th epoch without overfitting occurs during the training process. The model is trained for about 40 iterations and the network loss values gradually smooth out.

The mAP metric of the model is the primary standard of the detector, and the average accuracy mAP curve of the model in this paper is shown in Figure 13. It can be learned from the curve that as the number of iterations increases, there is an increasing number of the value of mAP model. After the model is trained to 120 epochs, the DS-YOLO model has the highest mAP value, remaining at 89%.

TABLE 4. Model training hyperparameter.

Hyperparameter	Set value	Meaning
Batch size	8	The number of trainings samples in a batch
Epochs	120	Total number of iterations
Width	416	The width of the input image
Height	416	The height of the input image
Channels	3	The number of channels to input the image
Optimizer	Adam	Optimizer types
Momentum	0.937	Momentum parameters in optimization methods
Lr decay type	Cos	Learning rate decline way methods
Weight decay	0.0005	Weight attenuation coefficient
Init lr	0.001	Initial maximum learning rate of the model
Min lr	0.00001	Minimum learning rate of model
Label smoothing	0	Label smoothing parameter

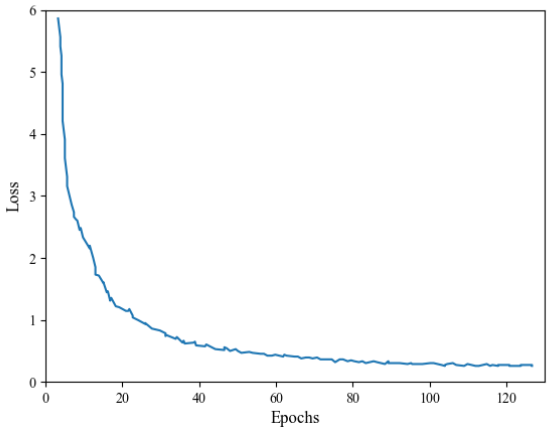


FIGURE 12. The curves of train loss and validation loss during the training process.

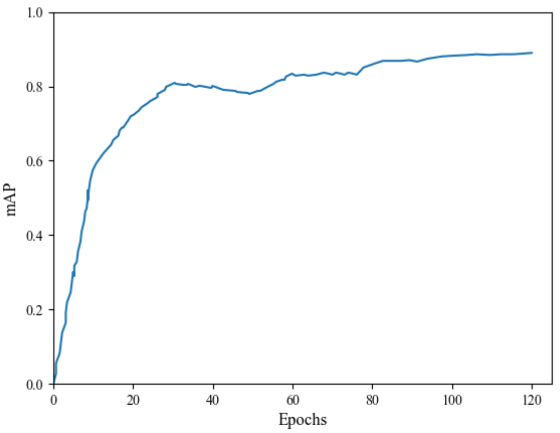


FIGURE 13. The curves of mAP during the training process.

V. EXPERIMENT

This section shows the performance testing of SCSS in real life applications, including the online recognition of three

types of anomalous behaviors, namely Fighting, Car accident and Fall. As some of the target behaviors such as Car Accident are difficult to capture in real life, this section evaluates the SCSS by identifying these anomalous behaviors in two parts: the model test and the live online recognition test, which have been captured from the video footage.

A. MODEL ON-LINE ACCURACY TESTING

A total of around 400 test samples of normal and three abnormal behaviors were collected in the video screen test session, with 100 test samples of each action included. The test experiments were implemented on SCSS through Pytorch, a deep learning framework, and the GPU used was a 128 Core Maxwell.

The precision rate describes how well our model performs in predicting the positive category. It is calculated by dividing the number of true positives (TPs) by the sum of TPs and false positives (FPs). Recall is calculated by dividing the number of true positives by the sum of the number of TPs and false negatives. Precision and Recall are defined in (10) and (11).

$$Precision = \frac{TPs}{TPs + FPs} \tag{10}$$

$$Recall = \frac{TPs}{TPs + FNs} \tag{11}$$

The overall classification accuracy of our model was calculated as the ratio of correctly predicted observations (that is the sum of TPs and true negatives (TNs)) to the total observations (that is the sum of TPs, FPs, FNs and TNs), defined as (12), the Confusion Matrix of this model is given in Table 5. The DS-YOLO model test results are shown in Table 6 and the overall accuracy can be achieved at over 89.78%, which proves that the system can accurately identify different abnormal behaviors, and the detection time is around 1.2 seconds.

$$Accuracy = \frac{TPs + TNs}{TPs + FPs + FNs + TNs} \tag{12}$$

TABLE 5. Confusion matrix.

True label	Model prediction	
	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

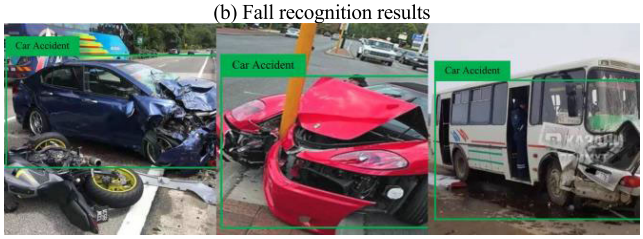
TABLE 6. Model online test results.

Behavior type	Recall rate	Recall	AP
Normal	0.8283	89.37	0.8712
Fight	0.9333	89.88	0.8932
Fall	0.7500	96.25	0.9012
Car Accidents	0.7759	90.01	0.9000

In the online testing section, we compared the algorithms with the YOLO series. the detection precision of our proposed

(a) Fighting recognition results

(b) Fall recognition results



(c) Car Accident recognition results

FIGURE 14. Volunteer test abnormal behaviors recognition in different scenarios.

DS-YOLO is compared with the algorithms of YOLO for fight, fall and car accident respectively in Table 7.

TABLE 7. The SCSS recognition results.

Model	Behavior type	AP	mAP (%)
YOLO	Fight	0.8593	86.04
	Fall	0.8175	
	Car Accident	0.9045	
DS-YOLO	Fight	0.8932	89.78
	Fall	0.9012	
	Car Accident	0.9000	

As can be seen from Table 4. Compared with the YOLO series models, the model combining YOLOv4 and the DeepSORT shows an improvement in accuracy, with mAP increasing from 86.04% to 89.78%. This model achieves competitive recognition accuracy.

B. LIVE ONLINE RECOGNITION TEST

This section describes the SCSS performance tests in actual use. All subjects were fully informed of the purpose of the experiment and signed a voluntary consent form.

Field tests were done in public places in the city on three abnormal behaviors: Fight, Car Accident and Fall. As the above behaviors are difficult to capture in real time in everyday life, five volunteers, three males and two females, between 160 cm and 185 cm in height were selected to simulate the actions of fights and falls. The validation data consisted of 700 test samples, 300 each of fights and falls, from urban public places, shopping malls and roadways.

As the IMX219 captured video under different weather conditions may have different phenomena in terms of color, aperture, exposure and other parameters. To ensure the stability of SCSS under different test environments, we conducted tests in the morning, midday and evening under sunny, cloudy and cloudy weather conditions to ensure the universality of SCSS under different weather conditions.

As shown in Figure 14, the three anomalous behaviors of Fighting, Car Accident and Fall can be quickly identified by SCSS, whether in public places or isolated suburban areas, even in poor weather conditions or in the presence of obstructions.

The SCSS recognition results are shown in Table 8.

TABLE 8. The SCSS recognition results.

Behavior type	Weather	Number of samples	Missed detection/ False detection	Accuracy (%)
Fight	Sunny	100	7	89.00
	Cloudy	100	12	
	Overcast	100	13	
Fall	Sunny	100	9	90.33
	Cloudy	100	10	
	Overcast	100	10	
Car Accident	Sunny	40	2	90.00
	Cloudy	30	4	
	Overcast	30	4	

Through field tests, it was found that the combination of YOLOv4 and DeepSORT could meet the recognition requirements in complex environments very well, and the recognition accuracy would be higher in places with less foot traffic than in obscured situations due to less obscuration, but in places with dense foot traffic, the overall recognition accuracy could still be maintained at 89.78%.

VI. CONCLUSION AND FUTURE RESEARCH

This paper introduces a smart city security system for online identification of public places, with two main functions: abnormal behavior detection and real-time monitoring and alarming, which can simultaneously complete online detection of three abnormal behaviors, Fight, Fall and Car accident, aiming to contribute to social security and the construction of ''smart cities''.

To meet the demand for real-time target detection in intelligent security systems, the system uses the lightweight target detection algorithm YOLOv4 for feature extraction and target detection, combined with the DeepSORT algorithm to match the image target in the next frame to achieve the tracking effect. The DS-YOLO algorithm deployed on Jetson Nano can realize real-time online monitoring of abnormal behaviors, which can effectively improve the efficiency of handling abnormal events. The system uploads system monitoring information to the cloud as well as to the client via GPS and WIFI modules. Users can view the detected abnormal behavior in the client and choose to alarm or view the historical abnormal records. Besides, a UPS is also installed to protect the system's power supply. Experiments have demonstrated that the model has a lower target miss detection rate compared to the YOLO series models, and the overall accuracy of the system has improved to 89.78% while ensuring real-time performance, which is a 3.74% improvement compared to the YOLO series. Compared with other mainstream algorithms, the method has fast operating speed and high accuracy.

In future work, we will be devoted to expand the dataset to include more types of anomalous behavior under more environmental conditions. In terms of algorithms and models, we will improve the model architecture, using more powerful embedded hardware platforms and optimize the anomalous behavior detection model to achieve higher recognition accuracy and faster recognition speed. At the same time, in the aspect of the shell mechanical design, we will strengthen the waterproof characteristics of the product, thus ensuring the safety and stability of the application.

ACKNOWLEDGMENT

(Kun Xia and Lingxiang Zhang are co-first authors.)

REFERENCES

[1] J.-Y. Yu, Y. Kim, and Y.-G. Kim, ''Intelligent video data security: A survey and open challenges,'' *IEEE Access*, vol. 9, pp. 26948–26967, 2021.
[2] Z. Dong, J. Wei, X. Chen, and P. Zheng, ''Face detection in security monitoring based on artificial intelligence video retrieval technology,'' *IEEE Access*, vol. 8, pp. 63421–63433, 2020.
[3] Y. Ge, S. Lin, Y. Zhang, Z. Li, H. Cheng, J. Dong, S. Shao, J. Zhang, X. Qi, and Z. Wu, ''Tracking and counting of tomato at different growth period using an improving YOLO-deepsort network for inspection robot,'' *Machines*, vol. 10, no. 6, p. 489, Jun. 2022.
[4] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, ''A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets,'' *IEEE Access*, vol. 7, pp. 170457–170473, 2019.
[5] Z. Sun, J. Sun, and X. Li, ''Research on video quality diagnosis technology based on artificial intelligence and Internet of Things,'' *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–6, Dec. 2021.
[6] A. A. Ahmed and M. Echi, ''Hawk-eye: An AI-powered threat detector for intelligent surveillance cameras,'' *IEEE Access*, vol. 9, pp. 63283–63293, 2021.
[7] F. Wang, J. Qiao, L. Li, Y. Liu, and L. Wei, ''Scene recognition of road traffic accident based on an improved faster R-CNN algorithm,'' *Int. J. Crashworthiness*, vol. 27, no. 5, pp. 1428–1432, Sep. 2022.
[8] Q. Zhang, X. Chang, and S. B. Bian, ''Vehicle-damage-detection segmentation algorithm based on improved mask RCNN,'' *IEEE Access*, vol. 8, pp. 6997–7004, 2020.
[9] J. Fang, J. Qiao, J. Bai, H. Yu, and J. Xue, ''Traffic accident detection via self-supervised consistency learning in driving scenarios,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9601–9614, Jul. 2022.
[10] D. Tian, C. Zhang, X. Duan, and X. Wang, ''An automatic car accident detection method based on cooperative vehicle infrastructure systems,'' *IEEE Access*, vol. 7, pp. 127453–127463, 2019.
[11] F. He, ''Intelligent video surveillance technology in intelligent transportation,'' *J. Adv. Transp.*, vol. 2020, pp. 1–10, Nov. 2020.
[12] B.-S. Lin, T. Yu, C.-W. Peng, C.-H. Lin, H.-K. Hsu, I.-J. Lee, and Z. Zhang, ''Fall detection system with artificial intelligence-based edge computing,'' *IEEE Access*, vol. 10, pp. 4328–4339, 2022.
[13] S. Raghavendra, S. K. Abhilash, V. M. Nookala, and S. Kaliraj, ''Efficient deep learning approach to recognize person attributes by using hybrid transformers for surveillance scenarios,'' *IEEE Access*, vol. 11, pp. 10881–10893, 2023.
[14] B. A. Holla, M. M. M. Pai, U. Verma, and R. M. Pai, ''Enhanced vehicle re-identification for smart city applications using zone specific surveillance,'' *IEEE Access*, vol. 11, pp. 29234–29249, 2023.
[15] Z. Yu, J. Liu, M. Yang, Y. Cheng, J. Hu, and X. Li, ''An elderly fall detection method based on federated learning and extreme learning machine (Fed-ELM),'' *IEEE Access*, vol. 10, pp. 130816–130824, 2022.

[16] S. Sun, Y. Liu, and L. Mao, "Multi-view learning for visual violence recognition with maximum entropy discrimination and deep features," *Inf. Fusion*, vol. 50, pp. 43–53, Oct. 2019.
[17] F. U. M. Ullah, M. S. Obaidat, K. Muhammad, A. Ullah, S. W. Baik, F. Cuzzolin, J. J. P. C. Rodrigues, and V. H. C. De Albuquerque, "An intelligent system for complex violence pattern analysis and detection," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 10400–10422, Dec. 2022.
[18] L. Ye, T. Liu, T. Han, H. Ferdinando, T. Seppänen, and E. Alasaarela, "Campus violence detection based on artificial intelligent interpretation of surveillance video sequences," *Remote Sens.*, vol. 13, no. 4, p. 628, Feb. 2021.
[19] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1002–1014, Apr. 2018.
[20] H. Cui, Z. Wei, P. Zhang, and D. Zhang, "A multiple granular cascaded model of object tracking under surveillance videos," in *Proc. Int. Conf. Algorithms, Comput. Artif. Intell.*, Dec. 2018, pp. 1–8.
[21] A. B. Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Exp. Syst. Appl.*, vol. 91, pp. 480–491, Jan. 2018.
[22] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for IoT systems using probabilistic image encryption," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3679–3689, Aug. 2018.
[23] L. Zhou, Z. Qiu, and Y. He, "Application of WeChat mini-program and Wi-Fi SoC in agricultural IoT: A low-cost greenhouse monitoring system," *Trans. ASABE*, vol. 63, no. 2, pp. 325–337, 2020.
[24] X. Ma, X. Lu, Y. Huang, X. Yang, Z. Xu, G. Mo, Y. Ren, and L. Li, "An advanced chicken face detection network based on GAN and MAE," *Animals*, vol. 12, no. 21, p. 3055, Nov. 2022.
[25] X. Zhang, C. Xuan, Y. Ma, H. Su, and M. Zhang, "Biometric facial identification using attention module optimized YOLOv4 for sheep," *Comput. Electron. Agricult.*, vol. 203, Dec. 2022, Art. no. 107452.
[26] D. Liu, J. Liu, and P. Yuan, "Lightweight prohibited item detection method based on YOLOv4 for X-ray security inspection," *Appl. Opt.*, vol. 61, no. 28, pp. 8454–8461, 2022.
[27] S. Tu, Q. Zeng, Y. Liang, X. Liu, L. Huang, S. Weng, and Q. Huang, "Automated behavior recognition and tracking of group-housed pigs with an improved DeepSORT method," *Agriculture*, vol. 12, no. 11, p. 1907, Nov. 2022.
[28] G. Zhang, J. Yin, P. Deng, Y. Sun, L. Zhou, and K. Zhang, "Achieving adaptive visual multi-object tracking with unscented Kalman filter," *Sensors*, vol. 22, no. 23, p. 9106, Nov. 2022.
[29] C. Zhang, Z. Tang, M. Zhang, B. Wang, and L. Hou, "Developing a more reliable aerial photography-based method for acquiring freeway traffic data," *Remote Sens.*, vol. 14, no. 9, p. 2202, May 2022.
[30] C.-J. Liu and T.-N. Lin, "DET: Depth-enhanced tracker to mitigate severe occlusion and homogeneous appearance problems for indoor multiple-object tracking," *IEEE Access*, vol. 10, pp. 8287–8304, 2022.

KUN XIA received the B.Eng. degree in industrial automation and the Ph.D. degree in power electronics and power drives from the Hefei University of Technology (HFUT), Hefei, China, in 2002 and 2007, respectively. From 2007 to 2011, he was a Lecturer with the University of Shanghai for Science and Technology (USST), Shanghai, China. From 2011 to 2019, he was an Associate Professor and the Department Head with the Department of Electrical Engineering, USST. From 2015 to 2016, he was also a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. From 2020 to 2022, he was a Professor and the Vice President with the College of Innovation and Entrepreneurship, USST. Since 2022, he has been a Professor and the Vice President with the College of Mechanical Engineering, USST. He has been in charge of more than 50 research projects from the government and companies and published more than 80 papers. His research interests include motor and motor control and new energy application. He won the Third Prize of the Scientific and Technological Progress Award of Zhejiang Province, in 2017, and Shanghai, in 2020.

LINGXIANG ZHANG was born in China, in 1999. He received the B.Eng. degree from the Department of Electrical Engineering, Xichang University, Xichang, China, in 2021. He is currently pursuing the M.Eng. degree with the Department of Electrical Engineering, University of Shanghai for Science and Technology. His current research interests include deep learning and UAV photovoltaic inspection.

SHUAI YUAN received the B.Eng. degree in materials science and engineering from Southwest Jiaotong University (SWJTU), Chengdu, China, in 2015, and the M.Eng. degree in mechanical engineering from Tiangong University (TJPU), Tianjin, China, in 2018. From 2018 to 2020, she was an Assistant Researcher with the Tianjin Sino-German University of Applied Sciences (TSGUAS), Tianjin. Since 2020, she has been an Assistant Experimentalist with the University of Shanghai for Science and Technology (USST), Shanghai, China. She has been in charge of four research projects from the government and companies and published seven papers. Her research interests include mechanical engineering and innovation and entrepreneurship.

YANG LOU was born in China, in 1999. He received the B.Eng. degree from the Department of Electrical Engineering and Its Automation, University of Shanghai for Science and Technology, Shanghai, China, in 2021. He is currently pursuing the M.Eng. degree with the Department of Electrical Engineering, University of Shanghai for Science and Technology. His current research interests include sensor measurement systems and embedded chips.