Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Highly secure edge-intelligent electric motorcycle management system for elevators

Zongwei Zhu[1], Jing Cao[1]*, Tiancheng Hao[1], Wenjie Zhai[1], Bin Sun[2], Gangyong Jia[3] and Ming Li[4]

Abstract

Because of their portability, electric motorcycles are usually pushed into elevators by residents and charged in the home, which has serious safety risks. Traditional manual-based methods to manage this behavior have poor monitoring effects and high costs. As for automatic management systems using artificial intelligence (AI), the deployment method matters. Cloud-based deployment methods have the disadvantages of high latency, high risk of privacy leakage, and heavy network transmission loads. In this paper, we propose a highly secure edge-intelligent electric motorcycle management system for elevators. By using edge-based deployment method, the monitor pictures are processed locally without being uploaded to the cloud, which can effectively resist network attacks and prevent residents' private data from being leaked. To improve the system security, we fully analyze the challenges faced in the application scenarios and introduce security threat identification (STI-1H8) model to identify the security threats. In addition, we propose several data enhancement methods to improve the system recognition accuracy. Experimental results show that our system can achieve a high recall rate of 0.82. By using data enhancement and data mixing strategies, it can reduce the misjudgment rate by 0.35. Moreover, compared to cloud computing, our edge-based method can reduce the latency by 19.6%, meeting real-time requirements.

Keywords: Edge computing, Image processing, Single shot multibox detector, Data privacy

Introduction

Due to their high performance and ease of use, electric motorcycles are a common means of transportation for people around the world. For reasons such as convenience, saving money, and antitheft measures, residents often take electric motorcycles home and charge them indoors by using elevators. However, this method poses serious safety risks: if a fire in a confined space is caused by improper charging, the toxic gas produced by the combustion of the large-capacity battery in an electric motorcycle can kill hundreds of people within 90 seconds. According to statistics, there are approximately 2000 electric motorcycle fire accidents in China each year, and more than half of them occur during charging [1]. In 2018, China Public Security Bureau issued an emergency notice [2], prohibiting residents from parking and charging electric motorcycles in elevators, corridors, and rooms.

Therefore, it is extremely important to implement effective management methods against the illegal behavior of residents taking electric motorcycles upstairs through elevators. Manual-based methods currently used by community managers have the disadvantages of low investigation rate, high labor intensity, and low inspection efficiency. In recent years, an increasing number of AI-based automatic monitoring systems in the field of public safety have been applied [3–5]. There are two main deployment methods for AI-based security monitoring systems: cloud-based and edge-based architecture. The cloud-based method transmits a large number of residents' sensitive data, such as their appearance, their movement in the elevator, their

*Correspondence: congjia@mail.ustc.edu.cn
[1] Suzhou Research Institute, University of Science and Technology of China, Renai Road, Suzhou, China
Full list of author information is available at the end of the article

room number, etc., to the cloud through the Internet. If attacked, it will cause the leakage of residents' privacy. In addition, as the amount of data collected by terminal devices continues to increase, cloud-based methods bring massive data transmission pressure to the network, which cannot meet the real-time requirements of the safety monitoring system.

Compared with cloud computing, edge computing shows excellent performance in reducing communication delays [6, 7], alleviating transmission loads [8] and preventing user privacy leakages [9, 10]. However, there still exist several security threats. First and foremost, the computing power of edge devices is so poor [11] that it is impossible to build a complete security defense system. Second, the operating systems and communication protocols of edge devices are heterogeneous [12], making it difficult to design a uniform set of security rules for edge devices, which poses great challenges for system protection and management. Finally, edge devices have lower access rights than cloud servers and are vulnerable to attackers obtaining private data. These limitations make some attacks that are ineffective against cloud servers with high access control rights and complete security defense systems pose a great threat to edge devices. In the application scenario of this paper, if the electric motorcycle management system for elevators encounters attack, causing the opening and closing of the elevator door to be out of control, it will pose a great threat to the personal safety of residents. As far as we know, there is currently no solution that can use the characteristics of an electric motorcycle detection system in an elevator to identify safety threats. Therefore, it is necessary to design a simple and efficient safety identification system to fully exploit the system characteristics and ensure the safety of the system.

In recent years, many researchers have conducted in-depth research on security protection for edge computing from different aspects [6, 10]. From the aspect of risk analysis, Roman et al. [13] conducted a security analysis of several common edge-based architectures and introduced a universal security protection system. Xin Tong et al. [14] used threat trees to carry out threat modeling. These works provide a great guidance for the design of the system security protection scheme of this paper. In terms of risk identification, frequently used machine learning and deep learning methods [15, 16] are difficult to perform well in edge devices with limited resources.

In this paper, we propose a highly secure edge-based automatic electric motorcycle management system for elevators. The system runs on a *Cambricon 1H8* [17] edge-intelligent device with a camera installed in the elevator. To improve system security, we introduce a privacy security model, *STI-1H8*, combining the application scenarios of the system proposed in this paper. In addition, to improve the system recognition accuracy, we fully analyze the challenges faced in the application scenarios and propose several data enhancement methods. Experimental results show that our system can achieve a high recall rate of 0.82. Moreover, by using data enhancement and data mixing strategies, it can reduce the misjudgment rate by 0.35 compared to using the original dataset. Simulated attack experiment shows that the *STI-1H8* model can recognise 100% of the application layer attacks, 81% of the network layer attacks, and 84% of the perception layer attacks. Comparative experiments show that the edge computing solution has a latency 19.6% lower than that of cloud computing. Moreover, compared with the mainstream electric motorcycle detection system *Kediou*, the power consumption of our system is only half that of *Kediou*, but its recall rate is improved by approximately 22%.

The contributions of this article are as follows.

First, compared to the commercial electric motorcycle detection system, the system proposed in this paper uses a *1H8* edge-intelligent device with high energy efficiency to reduce the energy consumption. At the same time, it analyzes the challenges faced in the detection process combined with the application scenarios in detail and can achieve a higher energy efficiency ratio based on the premise of ensuring recognition accuracy.

Second, by using the *STI-1H8* privacy security model, the system can identify multiple security risks. For each risk, by combining the industry's mature defense solutions, the system security can be greatly improved. In addition, the security modeling provided by this system can also support the security protection of other edge smart devices.

The rest of this paper is arranged as follows. "System design" describes the system design and introduces the safety modeling method and data enhancement method. "Experiment" illustrates the experimental results. "Related work" introduces the related work. Finally, "Conclusion" sections concludes the paper.

System design

In this paper, we propose an edge-based automatic electric motorcycle management system for elevators, which aims to ensure the timeliness of data processing and real-time feedback results. The device is installed on the ceiling of the elevator, and the camera mounted on it can obtain pictures in real time. If an electric motorcycle enters the elevator, the device will trigger a voice alarm and trigger the relay to keep the elevator door open until the resident pushes the electric motorcycle out of the elevator.

Overview of the system architecture

As shown in Fig. 1, the system consists of six modules, in which the *video processing module*, the *image analysis and*

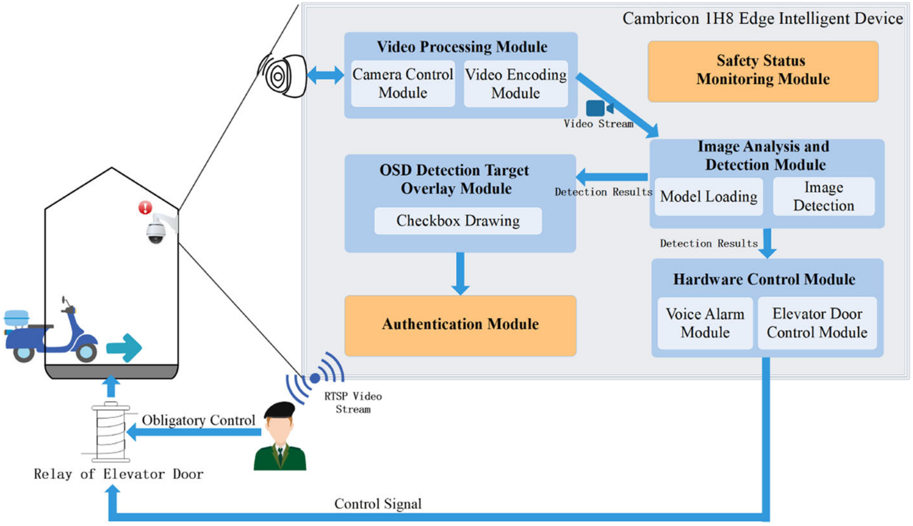


Fig. 1 System architecture

detection module, the *OSD detection target overlay module* and the *hardware control module* are the main functional modules of the system. The *authentication module* sends the Real Time Streaming Protocol (RTSP) video stream containing the recognition results to community security managers in real time through the network and receives instructions from the managers. The *safety status monitoring module* monitors system parameters in real time, identifies the system's security status, and notifies community security managers when the system is under attack. The mechanism to identify the security status of the system is introduced in the next section. The functions of the four main functional modules are introduced as follows.

Video processing module

The main responsibility of *the video processing module* is to call the camera module to obtain video frames and then pass the acquired pictures to the *image analysis and detection module*. During operation, it first checks whether the *video encoding module* is running successfully. Then, the *camera control module* reads the device address, encoding type and resolution of camera module and calls the camera. The pixel format is set to Luminance-Bandwidth-Chrominance (YUV) 420. Next, the output frame rate of the encoder is set, and the Video Encoder (VENC) *video encoding module* is called to encode the video. In this system, a H.264 encoder with a high video compression rate is used as the video encoding and decoding protocol. Each video frame only needs to record the difference in the pixel value, brightness and color temperature from the previous frame. This encoding method can effectively reduce the encoding and decoding rate of H.264 encoder, thereby reducing the transmission delay and meeting the real-time requirements of the electric motorcycle management system.

Image analysis and detection module

The *image analysis and detection module* is the core module of the electric motorcycle management system. When the picture is transferred to the Dynamic Random Access Memory (DRAM) of the *Cambricon 1H8* edge-intelligent device, the program will call the neural network model embedded in the *1H8* device to process the picture and recognize the electric motorcycle in the picture. Then, it passes the position of the electric motorcycle to the *OSD Detection Target Overlay Module*. For the recognition frequency, the program recognizes a picture every 0.5 seconds. Only when the electric motorcycle target is recognized on three consecutive pictures is it considered that there is an electric motorcycle currently entering the elevator. Then, the recognition result is transmitted to the *Hardware Control Module* to execute the corresponding hardware control. It is worth noting that due to the limited computing power and storage of the *1H8* device, there are certain requirements for the complexity of the neural network model. In the following section, we will introduce the model selection procedure and several methods to improve the recognition accuracy in the specific application scenario.

OSD detection target overlay module

The *OSD detection target overlay module* draws a checkbox in the original picture to mark the recognized electric motorcycle target. Therefore, community security managers can use the RTSP video streaming client to view marked images in real time. The module uses On-Screen Display (OSD), an on-screen menu style adjustment method that displays characters, graphics and other information on the display terminal. The model detection results are then displayed synchronously using the marked images to achieve real-time detection.

Hardware control module

The *hardware control module* consists of the *voice alarm module* and the *elevator door control module*. After receiving the recognition result from the *image analysis and detection module*, if an electric motorcycle is entering the elevator, the *voice alarm module* will play a warning voice message to alert the passenger. In addition, the *elevator door control module* sends a control signal to the relay. Then, the relay outputs a high-level signal to activate the elevator door opening button to keep the elevator door open. After the *image analysis and detection module* confirms that there is no electric motorcycle in the elevator, the *elevator door control module* sends a control signal to the relay to close the elevator door.

System implementation

The pseudo code of the system is shown in Algorithm 1. When the program runs, it firstly starts remote control signal receiver, relay and voice alarming device through the corresponding device addresses. The remote control signal receiver generates a high level signal after receiving the obligatory shutdown control signal from the community security manager. If the system continuously recognizes ten high-level signals, it will control the relay to work and force the elevator doors to close. Otherwise, it is considered that the community security manager has not sent an obligatory shutdown control signal at this time, and the system enters the recognition phase.

In the recognition stage, if the voice alarming device has not been operated and the electric motorcycle has been recognized for three consecutive times, the system starts the voice alarming device and controls the relay to work, so that the elevator door is continuously opened. On the contrary, if the alarm is working at this time, and the electric motorcycle has not been recognized for three consecutive times, it is considered that the resident has pushed the electric motorcycle out of the elevator at this moment, then, the system closes the alarm and the elevator door.

Design of safety status monitoring module

In edge computing, the management of security is particularly important.

The European Telecommunications Standardization Institute (ETSI) divides the security protection strategy for edge computing into three layers [18]. The first layer is physical security, which uses sensors to detect the physical state of the node and discover abnormal equipment in time. The second layer is network security, which sets the security level of the edge computing node and performs regional isolation according to the level and strengthens the prevention against (D)DoS attacks. The third layer is application security, which mainly strengthens the prevention of security risks such as malicious and illegal access by third parties and the leakage of sensitive information.

Algorithm 1 Detection and System Control

```
Require: Path_ObligatoryControl, Path_relay, Path_alarm, Image
Ensure: Recognization Results
initialize:Set $Num\_gpio \leftarrow 0$, $Num\_Motorcycle \leftarrow 0$, $Num\_Not\_Motorcycle \leftarrow 0$, $ShakeNum \leftarrow 3$
Open the device file $fd\_oblctr$, $fd\_rly$, $fd\_alm$ through $Path\_ObligatoryControl$, $Path\_relay$ and $Path\_alarm$
for $i = 0$ to 10 do
    $READ(fd\_oblctr, value, 3)$
    if $ATOI(valus) = 1$ then
        $Num\_gpio \leftarrow Num\_gpio + 1$
    end if
end for
if $Num\_gpio < 10$ then
    $Num\_gpio \leftarrow 0;$
    if $AudioPlay = false$ then
        if $DetectMotorcycle(Image) = true$ then
            if $Num\_Motorcycle < ShakeNum$ then
                $Num\_Motorcycle \leftarrow Num\_Motorcycle + 1$
                $Num\_Not\_Motorcycle \leftarrow 0$
            else
                $Write(Path\_relay, “1'', 2)$
                $Write(Path\_alarm, “1'', 2)$
                Audio alarm working
                $Num\_Motorcycle \leftarrow 0$
                $Num\_Not\_Motorcycle \leftarrow 0$
            end if
        else
            if $Num\_Not\_Motorcycle < ShakeNum$ then
                $Num\_Not\_Motorcycle \leftarrow$
$Num\_Not\_Motorcycle + 1$
                $Num\_Motorcycle \leftarrow 0$
            else
                $Write(Path\_relay, “0'', 2)$
                $Write(Path\_alarm, “1'', 2)$
                $Num\_Motorcycle \leftarrow 0$
                $Num\_Not\_Motorcycle \leftarrow 0$
            end if
        end if
    end if
else
    $Write(Path\_relay, “0'', 2)$
    $Write(Path\_alarm, “1'', 2)$
    $Num\_gpio \leftarrow 0$
end if
Close the device file $fd\_oblctr$, $fd\_rly$ and $fd\_alm$
```

Based on the three-layer risk proposed by ETSI, we analyze the specific risks faced by the system in detail and divide the security risks faced by our system into the perception layer risk, network layer risk and application layer risk, as shown in Table 1. Then, we propose a security risk identification model. By monitoring real-time performance indicators of the system during operation, the model identifies whether the system is facing an attack. Combined with identified security threats, the system can be processed using industry-proven security solutions and notify the administrator to ensure the security of the system.

Table 1 Security risks faced by our system and the behavior of the system at risk

Layer	Attack behavior	System behavior
Perception layer	Physical attack. Attackers destroy and disassemble the smart camera, causing it to break	The system fails to operate normally, and all indicators drop to zero suddenly.
	Attackers illegally insert and remove the SD card, causing the system to crash.	
	Attackers use fake electric motorcycle images to simulate an attack, frequently initialize corresponding components, and consume system resources.	The proportion of the target area is quite different from the proportion of the target in the real scene.
Network layer	Flooding (Dos) attack, which causes excessive consumption of system resources through frequent attacks and cause the system fail to process normal requests in time.	The memory utilization, device power consumption and network bandwidth are greatly improved, and the CPU utilization spikes violently.
	Attackers attack the network components and illegally tamper with the network configuration.	Various performance indicators of the system increase.
Application layer	Illegal intrusion into the device and frequent initialization of the program module	Various performance indicators of the system increase.
	Illegal intrusion into equipment and malicious modification of the program data.	System operation is abnormal, and various performance indicators change significantly.

We observe the performance indicators of the system during normal operation, including CPU utilization, memory utilization, device power consumption, and network bandwidth. Through analysis, we find that ignoring the impact of device startup, under normal operating conditions, the memory utilization rate, device power consumption, and network bandwidth are around a fixed value. The changes in various system performance indicators with time will be given in the "Experiment" section. According to the changes in the system performance indicators and the proportion of the target area in the picture, a security threat identification model, *STI-1H8*, is established. *STI-1H8* is a recognition model to judge whether the system is being attacked. It consists of the following elements: *STI-1H8 (C, M, N, P, A)*, where

- **C** is the variance in CPU utilization. It is used to measure the fluctuations in CPU utilization. When certain attacks occur, the system's CPU utilization will be less stable than normal. In addition, DoS attacks can cause the CPU utilization rate to remain at full usage with less fluctuation.
- **M** is the average memory utilization value. When certain attacks occur, the memory utilization will increase or decrease by a significant amount compared to the normal state.
- **N** is the he average network bandwidth occupancy. When certain attacks occur, the network bandwidth occupancy will increase or decrease significantly compared to the normal state.
- **P** is the average power consumption of the device. When certain attacks occur, the power consumption of the device will increase or decrease significantly compared to the normal state.
- **A** is the proportion of the current identification target area. When a disguised data attack occurs, the proportion of its target area will be quite different from that of the target area.

The feature extraction method based on the K-means clustering algorithm [19] is common used for mining the characteristics of system performance indicators from their historical records [20]. In the normal state, each value of the input vector of the system should remain relatively stable; that is, it can fall into a spatial cluster. Therefore, we use the K-means algorithm to find the center of the cluster where all the normal samples are located. Normal samples are distributed around the cluster center. When an attack occurs, various indicators of the system will change abnormally, and the corresponding index vector will fall far away from the cluster center where the normal sample is located.

The workflow of the *STI-1H8* model is as follows:

1) **Model establishment stage:** Combined with the fluctuations in the system under normal operation, the K-means clustering algorithm is used to cluster the sample data to find the center point. Then, the maximum Euclidean distance A between the normal sample and the center of the cluster is found.
2) **Model judgment stage:** The Euclidean distance is calculated between the input vector and the center of the normal sample cluster. If it is greater than threshold A, the current vector is determined to be an abnormal vector, and the system is attacked.

Optimization of the recognition model

Compared with the cloud-based architecture, the edge-based architecture has the advantages of strong real-time detection, good privacy protection and low network load. However, the limited computing power and storage capacity of edge devices bring many restrictions to the choice of the network model. The model needs to meet the following basic requirements:

1) The model should be able to complete the recognition task quickly under limited computing resources and ensure a high recognition accuracy.
2) The model should not take up too much storage space on the edge device.

To meet the above requirements, we first select a suitable lightweight network model according to the computing characteristics of the *1H8* device. Then, we analyze the challenges faced when identifying electric motorcycles in elevators in depth. In response to these challenges, we use several data enhancement methods to improve the accuracy of the model.

Lightweight network model

In the field of target recognition, classic convolutional neural networks (CNNs), such as Visual Geometry Group (VGG) [21], MobileNet [22], and LeNet [23], need to combine specific network models, for example, single-shot multibox detectors (SSD) [24], to realize the recognition of the target. Our SSD is based on a feedforward CNN, which generates a series of bounding boxes of fixed size and scores the likelihood that the object in the box is identified as the appropriate category. As shown in Fig. 2, the network structure of the SSD consists of two parts: a basic network and a pyramid network. The basic network can obtain higher-resolution feature maps and can better identify small targets than the pyramid network. The pyramid network is a convolutional layer that follows the basic network and can obtain more small-size and low-resolution feature maps to better identify large targets than the basic network.

Due to the limitations of the computing power and storage space of edge devices, the correct choice of a lightweight CNN model is the key issue for maximizing system performance. MobileNet uses depthwise separable convolutions, which can reduce the number of parameters and increase the running speed. When trained on COCO train+val excluding 8k minival images and evaluated on minival, compared with VGG-SSD, MobileNet-SSD only loses 1.8 percent of the mean Average Precision (mAP), but uses 26.3 million fewer parameters. Since MobileNet-SSD is much smaller in calculation complexity and model size than VGG-SSD with small calculation accuracy loss, we chose the lightweight Mobilenet-SSD as the network architecture.

Considering the limited storage space, memory size and power consumption of edge devices, we perform 8-bit model quantification. Model quantification is a process that maps continuous floating-point weights to a finite number of discrete values of 8-bit integers with low precision loss. The quantified model is reduced to a quarter of the size of the original.

Privacy protection

In order to protect the privacy of residents from being leaked, the system proposed in this paper adopts an edge-based deployment method. We also propose the *STI-1H8* model to detect system security state. However, there is still a risk of monitoring data leakage. If attackers maliciously steal local data in SD card, or intercept RTSP video streams transmitted to administrators through the network, and then obtain residents' residence information through location inference, it will cause unnecessary trouble to residents.

To further protect residents' privacy, the picture are coded in system idle time. When the management system is not in the recognition cycle, for example, the elevator is on higher floor, the algorithm recognizes residents in the pictures stored in the SD card and codes their face. It's worth noting that the system idle time is much longer than the recognition time, so that all pictures can be coded in time. Then, the system overwrites the original pictures with the coded pictures. What's more, if the administrators request to view the real-time monitoring video stream, the system will encode the video while sending the RTSP video stream. Although this will cause greater pressure on the system in a short time, it guarantees the security of residents' private data. The coding results are as shown in Fig. 3.

Data enhancements

In actual application scenarios, affected by factors such as elevator specifications, the camera installation position,

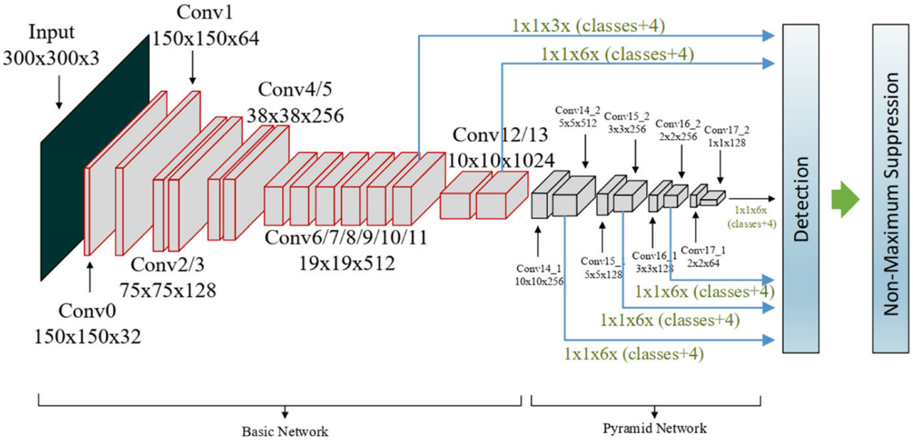


Fig. 2 The architeture of MobileNet-SSD

and light, the data collected directly by the camera often have problems such as the target being blocked or too small, insufficient brightness or overexposure, excessive noise, and images from a single angle, which can seriously affect the recognition accuracy. To reduce the influence of the above factors, we analyze 9963 pictures collected from elevators in depth, of which 18% are collected from commercial mixed-use apartment elevators and 82% are from residential building elevators, and propose the use data enhancement methods from the perspective of training the network models.

1) **Interference of Objects in the Elevator**
   There are a wide variety of objects in the elevator, including motorbikes, baby carriages, bicycles, etc., which have similar characteristics to electric motorcycles, and objects having nothing to do with electric motorcycles, such as water buckets, mops,

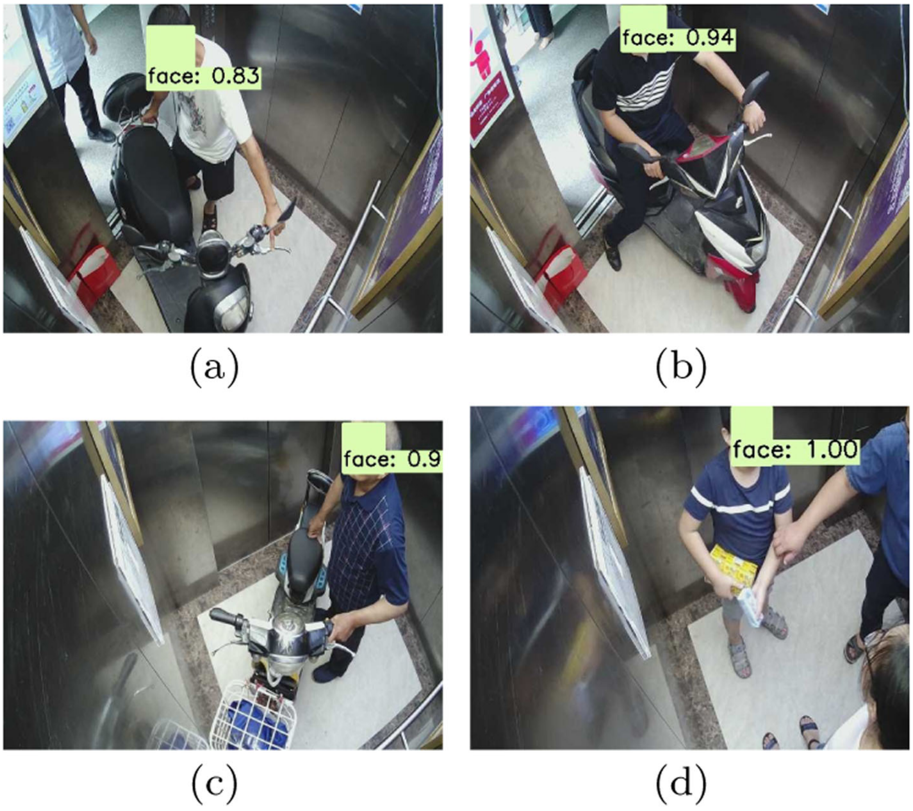


Fig. 3 Face coding to avoid location inference

trash cans, as well as building boards in the commercial mixed-use apartment elevators. Recognizing electric motorcycles from these wide-ranging objects, especially those that have similar characteristics to electric motorcycles, poses a massive challenge to the recognition model. To improve the accuracy of the recognition model and further improve its generalization ability, we propose a hybrid training method. We randomly select 2037 images from the VOC2007 dataset [25], mix them with the 9963 original image datasets, and scale all the pictures. All the pictures are proportionally resized to 300x300 to meet the requirements of the training model. In addition, to improve the model's ability to recognize objects that have similar characteristics to electric motorcycles, we increase the number of bicycles and motorbikes. After mixing the datasets and performing sample equalization, we obtain 721 motorbikes, 3417 bicycles, 2322 electric motorcycles, and 894 baby carriages.

2) **Difference in the Specifications fo the Elevators** In actual application scenarios, the specifications of different elevators vary greatly. As shown in Table 2, the load of the elevator varies from 320 kg to 2500 kg. As a result, the size of the elevator varies greatly, causing the difference in the installation height of the camera to be as much as 51.3 cm. Affected by these factors, the proportion of the same electric motorcycle target in the image collected by different elevator cameras varies greatly. It is worth noting that during recognition, the proportion of electric motorcycles in the picture changes dynamically. What we count in Table 2 is the average proportion of pictures collected by the camera in different elevators. Within the same pixel, the larger the proportion of the image occupied by the motorcycle, the clearer the details, and the richer the features, the higher the recognition accuracy of the model is. Therefore, to further improve the generalization ability of the model to identify electric motorcycle targets with different proportions in images, we propose a clipping data enhancement method. Clipping is the process of clipping a rectangular area around the edge of the image toward the center without destroying the original features. The final effect is equivalent to magnification, as shown in Fig. 4a and b.

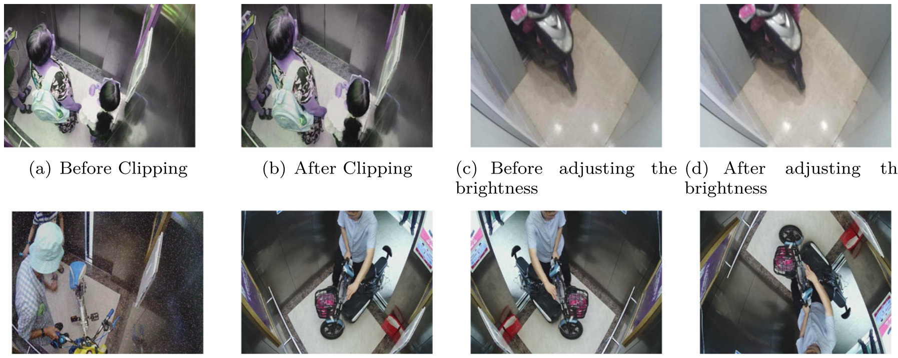
Table 2 Specifications of several example scenarios

	Lift Weight	Lighting Power	Camera Installation Height	Average Proportion of the Electric Motorcycle in the Images
Scenario 1	320 kg	3W	212.3cm	35.60%
Scenario 2	400 kg	3W	215.6cm	34.80%
Scenario 3	1200 kg	12W	221.4cm	21.20%
Scenario 4	1250 kg	12W	223.9cm	22.40%
Scenario 5	1600 kg	3W	238.7cm	21.60%
Scenario 6	2000 kg	15W	251.8cm	20.10%
Scenario 7	2500 kg	18W	263.6cm	18.60%

3) **Differences in the Elevator Interior Light Levels** The brightness of the light inside the elevator varies greatly throughout the day. Especially at night, the peak hours when residents go home, the light in the elevator is completely provided by the internal lamp. However, as shown in Table 2, the lighting power of different elevators varies between 3 W and 18 W, causing the brightness to vary greatly. In addition, the light in the elevator is also affected by factors such as the time of the day, the installation position in the elevator, the installation position of the lamp, and the type of lamp (such as tube lamp or disc lamp), which makes the brightness distribution extremely uneven in the collected datasets. In an environment where the brightness is too dark, the target features are unclear and cannot be captured well. Therefore, we propose a brightness adjustment data enhancement method to improve the low-brightness samples so that the model training process can better extract the features of the picture and improve the recognition accuracy under different brightness environments, as shown in Fig. 4c and d.

4) **Interference of the Picture Noise** During the peak electricity consumption period in residential buildings at night, the voltage of the power supply system varies greatly. The unstable voltage can add noise to the images collected by the camera. In addition, factors such as mechanical vibrations while the elevator is running can also add noise to the images. According to our analysis, due to the influence of noise and light, during the peak power consumption at night (that is, 19:00-22:00), the recognition accuracy of the model can be reduced by as much as 21.8% compared with that during the day. Considering the limited resources of the 1H8 device, running a denoising algorithm on it is not a good choice. To solve this problem, as shown in Fig. 4e, we propose the data enhancement method of adding noise. Several common types of noise are added to the original image to improve the adaptability of the neural network model.

5) **Camera Installation Position** Due to the different relative positions between the entrance of the corridor and the elevator, to fully



(a) Before Clipping

(b) After Clipping

(c) Before adjusting the brightness

(d) After adjusting the brightness

(e) Randomly adding Salt Pepper Noise.

(f) Before mirrored.

(g) After mirrored horizontally.

(h) After mirrored vertically.

Fig. 4 Data Enhancements. Pictures **a** and **b** show clipping. Pictures **c** and **d** show brightness adjustment. Picture **e** shows the addition of noise. Pictures **f** to **h** show mirroring

capture the electric motorcycle, the installation positions of the cameras in different elevators are different. For example, in Scenario 1, the entrance of the corridor is located on the right side of the elevator, so the camera needs to be installed in the upper-left corner of the elevator to better capture the images, while Scenario 2 is the opposite. Based on this analysis, it can be seen that the target to be recognized should have rotational invariance. To further improve the adaptability of the system in different scenarios, we propose a data enhancement method of mirror transformation to train the model. The target of the electric motorcycle to be identified can still be regarded as the same target after mirroring. In the practical application, we adopted horizontal mirroring and vertical mirroring. As shown in Fig. 4f to h, the horizontal (vertical) mirroring operation performs a mirror image exchange on the image centered on the central vertical (horizontal) axis.

Experiment

Experimental setting

Experimental environment

The system proposed in this paper runs offline on the *Cambricon 1H8* edge-intelligent device. As shown in Figs. 5 and 6, it is equipped with a camera, which can obtain image data in the elevator in real time. After obtaining the image data, the system transfers it to the internal DRAM. Then, it is transmitted to the central *Floating-Point Unit (FPU)* for processing, which is a dedicated processor for floating-point operations. The system is connected via network cables to a display viewable by community security managers, and the real-time detection results can be sent to the front-end display through the RTSP video stream.

The configuration of the training environment is shown in Table 3.

Training dataset

During the experiment, we used three different training datasets:

1) **Raw dataset**: 2000 raw images without data enhancement.
2) **Expanded dataset**: 6000 images by expanding the raw dataset using the four data enhancement methods proposed in this paper.
3) **Mixed dataset**: 16,413 images by mixing the VOC2007 [25] dataset and the expanded dataset.

Recognition effect evaluation metric

We select three parameters: the recall rate, misjudgment rate and omission rate as the evaluation metrics. S is the set of targets in all of the test sets, M is the set of electric motorcycle targets in the test set, N is the set of targets identified as electric motorcycles in the test results, and P is the intersection of M and N. Then, the target recall of the electric motorcycle is P/M, the misjudgment rate is $(N-P)/S$, and the omission rate is $(M-P)/M$.

The misjudgment rate is the core parameter to measure the system performance. If the system generates a misjudgment, triggers the voice alarm and prevents the elevator door from being closed when there is no electric motorcycle, it may cause an inconvenience or even danger to residents.

System performance verification

To evaluate system performance, we conduct experiments from two aspects: verification of security capability, and verification of the system's recognition effect after data enhancement.

Verification of the security capability

To generate the *STI-1H8* model, we collect the curves of CPU utilization, memory utilization, device power consumption, and network bandwidth when the system

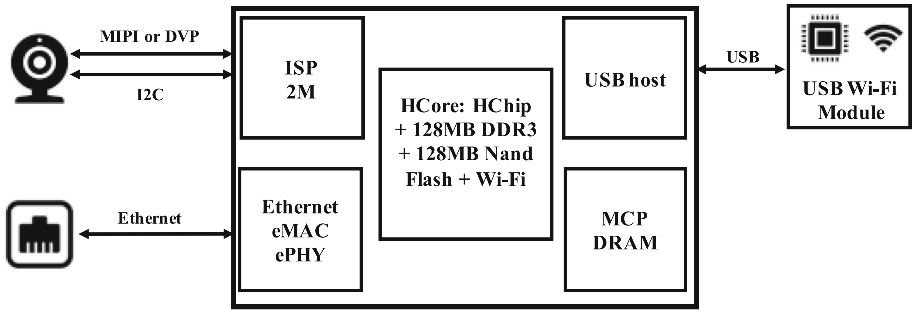


Fig. 5 Internal structure of 1H8

is steadily and safely operated for 132.5 s. As shown in Fig. 7, after the system is turned on, it takes about 13.5 s to start the program and load the data. Then, the system cyclically recognizes the electric motorcycles at a cycle of approximately 480 ms. To enhance the anti-interference ability, we perform feature modeling in intervals of 2 s and finally generate 48 sample *STI-1H8* vectors marked as safe. The power consumption and memory usage are highly correlated, with a correlation coefficient of 0.902, meaning that we should only choose one of them for modeling. Based on these safe STI-1H8 vectors, we take the average power consumption, the variance in CPU utilization and the average network bandwidth as the input of the K-means clustering algorithm.

After calculation, the coordinates of the center point in the cluster are (4.238, 25.789, 2.812), and the maximum Euclidean distance between all the safe vectors and the center point is 18.98. We take (4.238, 25.789, 2.812) as the center and 18.98 as the radius to draw the sphere. The range included in the sphere is the predicted normal area. However, even if all the monitored system indicators fall in the sphere, the system may still be subject to application layer attacks. This is because certain application layer attacks will not cause system indicators to fluctuate drastically. To increase system security, we analyze the fake image attacks in Table 1 that the application layer often suffers from. If all the system indicators are normal, the algorithm then checks the image proportion. Only when the proportion of the picture belongs to the range [0.186,0.60] will the system be deemed to be completely safe. The detection of other application layer attacks will be considered in our future work.

Then, to verify the validity of the *STI-1H8* model, we design 40 attack vectors to simulate various attacks that may be encountered during the operation of the system. We thoroughly analyze mass real-time data of the system and calculate the correlation coefficient between each index. After calculation, we find that the CPU usage rate is not significantly related to the other four indicators. The correlation coefficient between the network bandwidth and memory usage is only 0.0643. We also analyze the proportion of electric motorcycle targets in the picture in the dataset. As shown in Fig. 8, in the mixed dataset, the maximum target ratio reaches 60%. Therefore, if the recognized target accounted for more than 60% in the picture, the system is subjected to fake image attacks. Based on the above analysis, we propose the following design criteria of simulated attack vectors.

1) **Perception layer attack vector design**: The CPU utilization rate should be higher or lower than normal state by more than 60%. The power

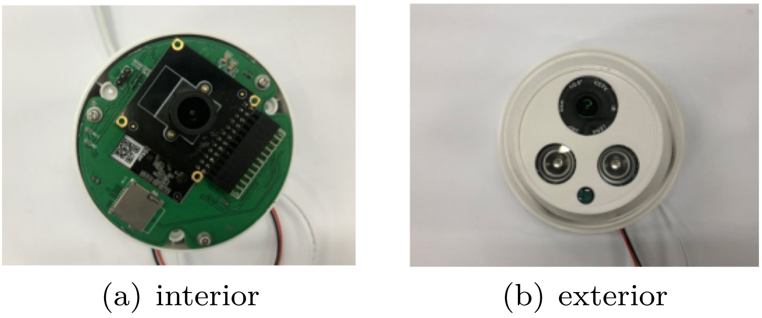

(a) interior (b) exterior

Fig. 6 The physical map of the system

Table 3 Configuration of training environment

GPU	Nvidia TITAN Xp
Deeplearning Framework	Caffe 1.0.0-rc3
CUDA version	10.1
Operating System	Ubuntu 16.04

consumption should be higher or lower than normal state by more than 40%.

2) **Network layer attack vector design**: Network bandwidth should be higher or lower than normal state by more than 30%.
3) **Application layer attack vector design**: The proportion of the recognized electric motorcycle targets in the picture should exceed 60%. In addition, to ensure the effectiveness of the attack vector, the other indicators should be normal.

To ensure the authenticity and effectiveness of the attack vector, all attack vectors should comply with the following basic principles.

1) The CPU usage should change independently of the remaining four indicators.
2) The network bandwidth should change independently of the remaining four indicators.
3) The power consumption and memory usage should change simultaneously.

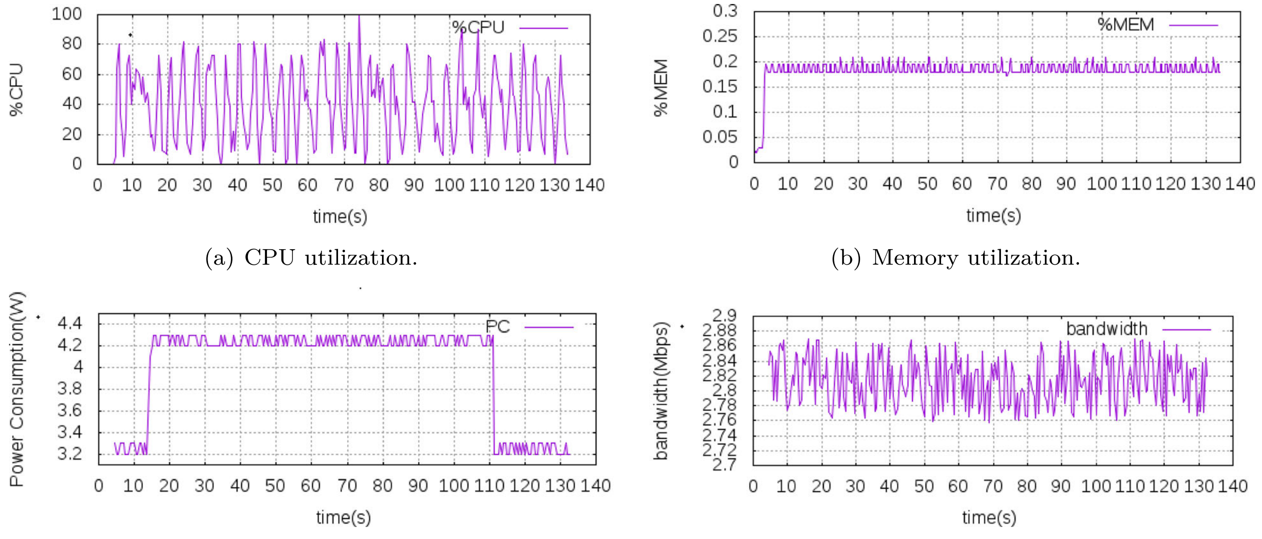
Considering attacks on the perception layer are the most harmful to the system performance, to test the robustness of the system, among the 40 simulated attack vectors, application layer attacks account for 7.5%, network layer attacks account for 27.5%, and perception layer attacks account for 65%, as shown in Fig. 9. Then, we use the *STI-1H8* model to identify the attack vector. The experimental results are shown in Fig. 9. The recognition rate of network layer attacks is 81%, and that of perception layer attacks is 84%. All application layer attacks are correctly identified. This is because the proportion of pictures in all application layer attack vectors is not in the range [0.186,0.60].

Verification of the effect of data enhancement

We train the MobileNet-SSD three times using different datasets to verify the misjudgment rate of the system. In the first training process, we use the raw dataset. In the second training process, we use the expanded dataset. In the third training process, we use the mixed dataset. These three training processes are performed with 30,000 iterations, and the confidence level is set to 0.7.

Table 4 shows the three evaluation metrics of the system. As we can see, in the third training process, after 30,000 iterations, the recall rate and the omission rate reach 0.82 and 0.17, respectively.

Figure 10 shows the change in the misjudgment rate during the three training processes. After 10,000 iterations, the model has still not been fit. As a result, using the original data as the training data can cause the model to fail to fully mine the target features and increase the misjudgment rate. The results of the second training and the third training show that using multiple data enhancements can effectively reduce the misjudgment rate, and using data enhancement and data mixing strategies can reduce the misjudgment rate by 0.35 compared to using the original dataset.



(a) CPU utilization.

(b) Memory utilization.

(c) Power consumption.

(d) Bandwidth.

Fig. 7 System performance indicators

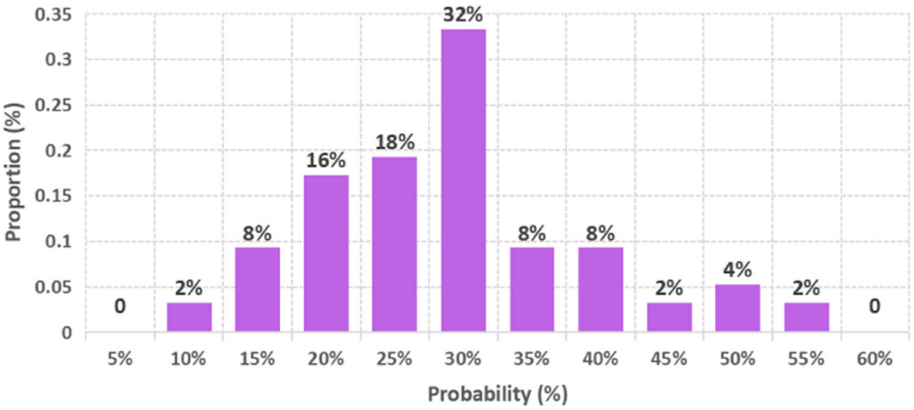


Fig. 8 Proportion distribution of the electric motorcycles

Therefore, in the following "Comparative experiment" section, we conduct our training process using the mixed dataset.

Comparative experiment

To verify the superiority of the system architecture adopted in this paper in terms of real-time performance, low power consumption, and low misjudgment rate, we conduct comparative experiments with cloud-based architecture, different recognition modes, and *Kediou*, another electric motorcycle detection system on the market, separately.

Comparison of edge computing and cloud computing

To verify the advantage of the low latency of the edge-based method we adopted, we perform detection using an electric motorcycle image separately in the *Cambricon 1H8* platform and the cloud computing platform and record the amount of data transmitted and time spent. Figure 11 shows the sequential differences between the two platforms.

The cloud computing process typically entails five steps: capturing the image, uploading it, processing it in the cloud, downloading the detection image and the result, and taking a control action by the camera device the according to the result. It should be noted that the cloud host hardware is an *Intel(R) Xeon(R)* CPU E5-2609 v4 @ 1.70 GHz, 64 GB RAM and a 500 GB solid state disk, the GPU is Nvidia TITAN Xp, and the network test environment is built with 200 Mbps bandwidth. An electric motorcycle detection RESTful API service is deployed on the Kubernetes cloud operating system. In our system based on edge computing, the electric motorcycle detection device with a 1H8 intelligent edge camera has the ability to process images; therefore, there is no need to download the result. After capturing and processing the

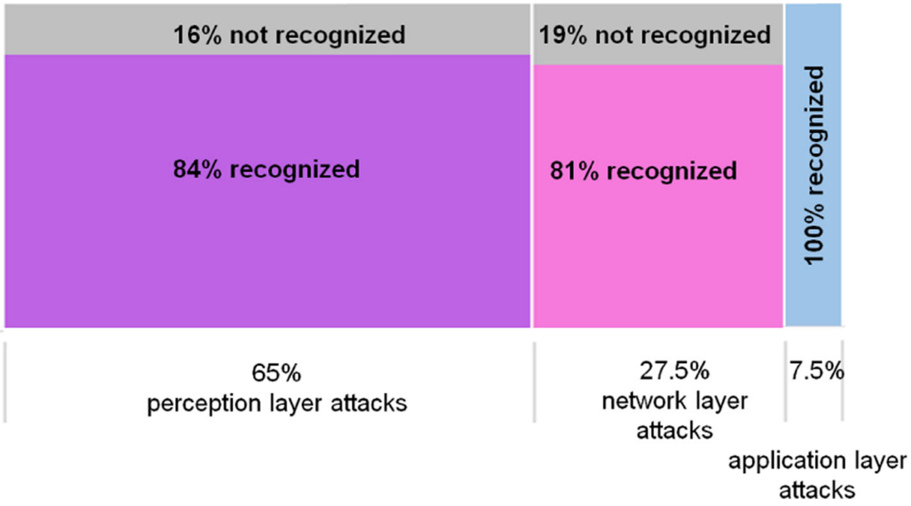


Fig. 9 Verification of the security capability

Table 4 Evaluation of the system

Training process	Number of iterations	Recall rate	Misjudgment rate	Omission rate
First training	10000	0.74	0.19	0.25
	20000	0.80	0.37	0.20
	30000	0.81	0.40	0.18
Second training	10000	0.85	0.25	0.14
	20000	0.84	0.20	0.15
	30000	0.84	0.20	0.15
Third training	10000	0.81	0.22	0.18
	20000	0.82	0.08	0.17
	30000	0.82	0.05	0.17

image, the electric motorcycle detection device takes a control action and subsequently uploads the detection image (context requests) to the cloud, and the result is returned.

As the gray part of Table 5 shows, the process of capturing the image by the electric motorcycle detection device response is defined as a response cycle. With cloud computing, capturing the image requires 40 ms. The uploading process takes 162 ms and includes packaging the detection image into JavaScript Object Notation (JSON) format and network delay. The processing time is 61 ms, which includes the API route analysis, image detection and result outputting. The download time for the detection image and the result takes 104 ms. This process includes network transmission and terminal data analysis. Finally, the electric motorcycle detection device spends 23 ms on receiving a command via a relay to take a control action. Therefore, the response cycle for cloud computing is 367 ms. In contrast, if edge computing is used, the network I/O time is significantly reduced, and the response cycle only includes capturing the image (40 ms), edge processing (232 ms), and elevator control (23 ms). Thus, the detection cycle only takes 295 ms. In summary, compared with cloud computing, edge computing reduced the latency by 19.6%.

Comparison of different recognition modes

There are two ways to design the recognition system: no matter which floor the elevator is currently in, it will cyclically detect whether there is an electric motorcycle, and, only detect the electric motorcycle on the first floor. To choose the optimal mode, we conduct a comparison experiment. Since the number of floors of buildings varies greatly, to simplify the experiment, we set the floor number as 5. During the experiment, the elevator starts at the first floor, stops at each floor and eventually reaches the fifth floor. The whole process takes 320s.

The experimental results show that although the former recognition mode can guarantee a higher inspection rate, the system is in a state of higher power consumption for a long time. As shown in the Fig. 12a, the power consumption of the device is between 4.2W and 4.3W for a long time, and the energy efficiency is relatively low.

In the actual application scenarios, the behavior of pushing electric motorcycles into elevators generally occurs on the first floor and the negative first floor (if this building has a negative first floor). Therefore, to make better trade-off between inspection rate and energy efficiency, we only recognize these floors. As shown in Fig. 12b, the system reads the floor information of the elevator from the

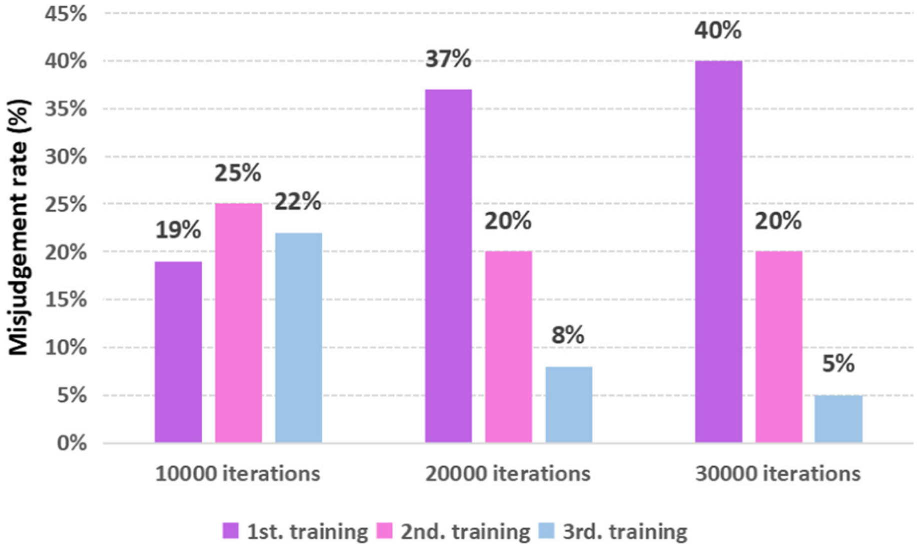


Fig. 10 Misjudgment rates of the three training processes

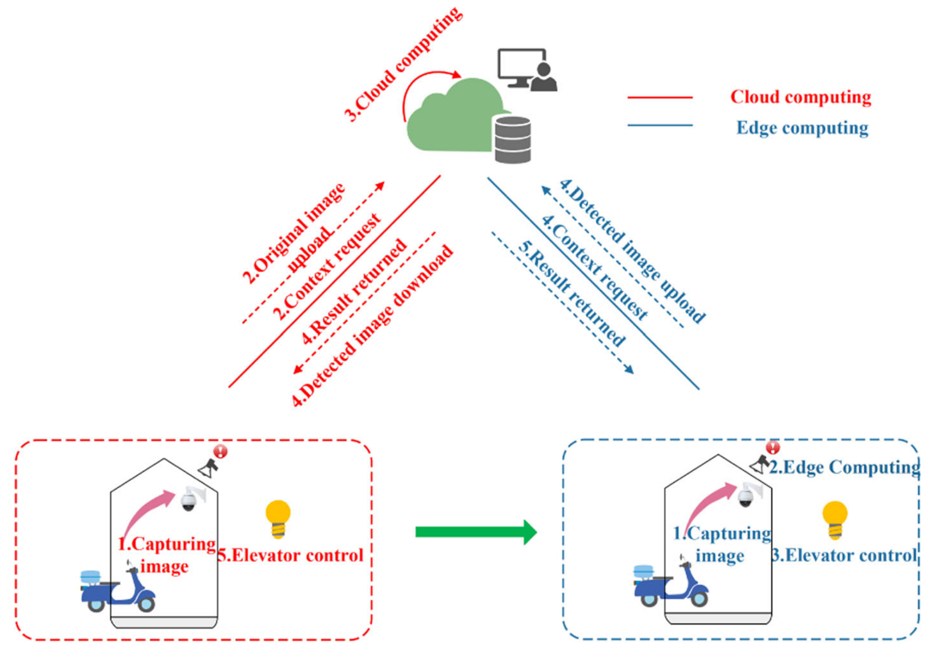


Fig. 11 Differiences between cloud computing and edge computing

elevator control system. When the elevator leaves the first floor, it stops recognition, and the power consumption of the device is between 3.2W and 3.3W, which greatly reduces the power consumption. In addition, avoiding long-term high-power operating states also reduces security risks such as device overheating.

Based on the above reasons, we choose to start the system recognition program only on the first floor or the negative first floor.

Table 5 The response times of edge computing and cloud computing

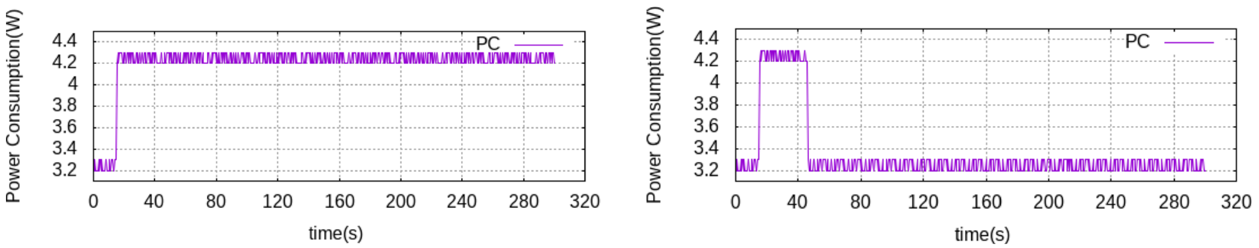
	Edge computing	**Cloud Computing**
Step1	Capturing the images	Capturing the images
	40 ms	40 ms
Step2	Edge computing	Uploading original images
	232 ms	162 ms
Step3	Controlling the elevator	Cloud computing
	23 ms	61 ms
Step4	Uploading the detected images	Returning the result
	162 ms	104 ms
Step5	Returning the result (no images)	Controlling the elevator
	64 ms	23 ms
Response cycle	295 ms	367 ms

Comparison of the 1H8 and Kediou

In this section, we compare another edge-intelligent electric motorcycle detection system on the market, KDO-PWT7102FDB-Y, which is developed by *Kediou* company based on the *Hi3516* chip. *Hi3516* is a professional high-end System On Chip (SOC) chip developed by HiSilicon for high-definition IPCamera product applications. Table 6 shows the comparative data of the hardware configuration and experimental results. In addition, the evaluation data are presented in two aspects:

1) Fig. 13a and b show that the *Kediou* device always misjudges some irrelevant objects during the recognition process.
2) Fig. 13c and d show that compared with the 1H8 device, under a confidence level of 0.65, the omission by the *Kediou* device is serious, and some targets cannot be identified.

In summary, through the comparison experiment before and after data enhancement, multiple data enhancement methods for the training samples have a positive effect on improving the accuracy of the model. By comparing the two methods of edge computing and cloud computing, we can verify that the proposed edge-based model can better solve the problem of transmission delay caused by cloud computing, making the system meet real-time requirements. By comparing our system with another mainstream system in the market, our system is better than the other system in terms of power consumption, accuracy and other indicators.



(a) Cycle recognition.

(b) Recognize only on the first floor.

Fig. 12 Comparison of different recognition modes

Related work

In the field of automatic safety inspection, AI-based methods have been widely used. As a computationally intensive task, AI is often combined with cloud computing. However, cloud-based architectures often face issues such as high latency and high network bandwidth consumption [26]. In contrast, edge-based systems have advantages in real-time tasks. However, it brings new problems to the security of the system and has higher requirements for deep learning models. In recent years, there have been many studies on improving the security of edge computing systems and improving the performance of edge-based target recognition. The related work is summarized as follows.

Security protection of edge computing

In recent years, many studies have been conducted on the security protection of edge computing.

Yinhao Xiao et al. [27] established the most advanced security attack and defense mechanisms in edge computing. By pointing out the root causes of edge computing security threats, they gave the corresponding solution. Roman et al. [13] conducted a security analysis of several common mobile edge paradigms and described a universal collaborative security protection system. Yuting Zhang et al. [28] proposed the importance of perception-layer security in Internet of Things (IoT) security and introduced the risks faced by the IoT perception layer and related security mechanisms. Similarly, based on the analysis of the security threats in the perception layer of the IoT, Xin Tong et al. [14] used threat trees to carry out threat modeling and described the consequences of major attack threats.

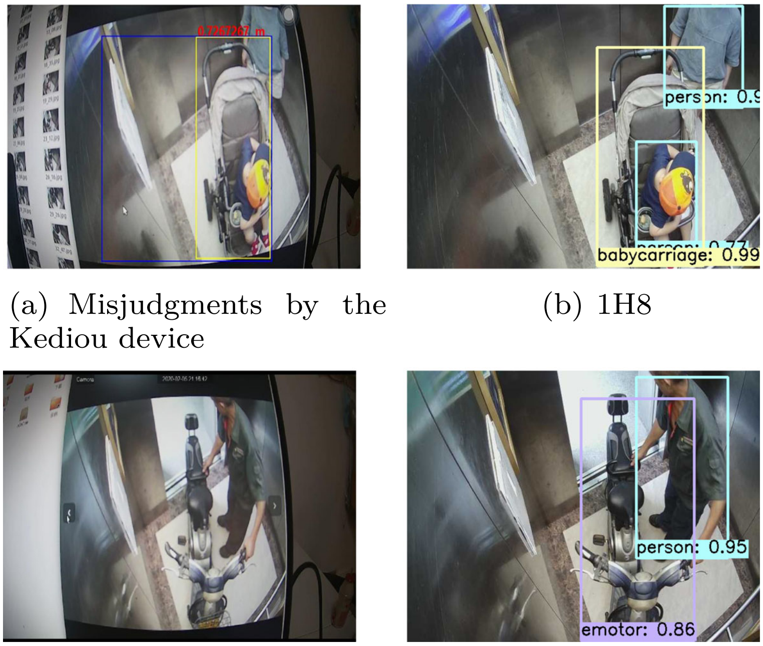
In terms of risk identification, machine learning and deep learning methods have already been used for more than a decade [15, 16]. With the development of computing power, increasingly more machine learning and deep learning methods are used to protect IoT systems. However, they require a large amount of security data to train the model. The monitoring system should be efficient and able to adapt to different safety environments [29]. However, due to the limitations of the edge device itself, many complex deep learning models are difficult to perform well. The study performed by Abdulaziz Aborujilah et al. [30] showed that the impact of network flooding attacks on the CPU and network bandwidth is very significant, and a series of experimental verifications were carried out in their research. These findings have greatly helped our research.

Implementation of a safety supervision algorithm

Cloud-based safety supervision systems have been widely used in recent years [31, 32]. However, they have many problems, such as long delays and privacy leakage. The development of edge computing can solve some of the problems existing in cloud computing.

Table 6 Comparison of the 1H8 and Kediou

	Parameter	**1H8** [17]	**Kediou**
Hardware configuration	Processor	1*H*8	*Hi*3516
	Image resolution	1920x1080	640x480
	Supports Wi-Fi?	Yes	No
Experiment results	Detection interval	Approximately 480 ms	Approximately 510 ms
	Power consumption	$\leq 5W$	$\leq 9W$
	Recall rate	$\geq 87\%$	$\geq 65\%$
	Operating temperature	$-20 \sim 50^{\circ}C$	$-20 \sim 60^{\circ}C$



(a) Misjudgments by the Kediou device

(b) 1H8

(c) Object omission of Kediou

(d) 1H8

Fig. 13 Comparison of misjudgments by the **a** Kediou and **b** 1H8 systems, as well as the comparison of the objects omitted by the **c** Kediou and **d** 1H8 systems

Christopher TJ Prentice et al. [33] proposed an intelligent office system based on edge computing. Compared with cloud computing, it can avoid the delay caused by the large amount of data uploaded to the server. In some specific application scenarios in the field of public security, the use of edge mobile devices can better meet the requirements of application scenarios, such as real-time detection of face information and alerting the police to the criminal records of dangerous persons encountered [34]. In addition, Dany-yuan found that 89% of Wireless-Fidelity (Wi-Fi) hotspots are unsafe, and massive personal privacy data are included in the real-time transmission of large amounts of data. It has been shown that edge computing can protect users' privacy better than a full cloud computing model [35]. At the same time, studies showed that edge computing combined with lightweight models can perform satisfactorily in human body detection, behavior recognition and intelligent monitoring [36]. Therefore, with the continuous improvement in processor performance, in some application scenarios, services based on edge computing have the advantages of low latency and high security compared with cloud computing.

Data enhancement

Data enhancement is an important way to improve the accuracy of models. Researchers performed random transformation, rotation translation, scaling and other data enhancement methods on skin lesion image samples to obtain an expanded dataset [37]. The experimental results showed that after the data enhancement processing, the mAP increased by an average of 3%. The experiments showed that the addition of augmented data can enhance the ability of neural networks to detect malignant skin lesions. In addition, the single-stage target detection algorithm is more sensitive to data enhancement methods than other methods. When the classic SSD algorithm is enhanced on the VOC2007 dataset, and the model mAP is improved by 6.7% [38]. Similarly, in a face recognition study, a single-stage target detection algorithm called Single Shot Scale-invariant Face Detector (S3FD) was used, and data enhancement methods such as random clipping and scaling were performed on the image samples. The experimental results also achieved a higher recognition accuracy with the addition of the enhanced data than with the original data [38]. Mateusz et al. [36] investigated the effects of sample imbalance on classification through three basic datasets, namely, the MNIST, CIFAR-10 and ImageNet dataset, which proved that sample imbalance inhibits classification performance. In summary, it is widely accepted in industry that data enhancement is an effective method to improve the accuracy of models.

Conclusion

In this paper, to solve the problems of cloud-based methods such as high network transmission pressure, poor real-time performance, and high risk of leaking residents' private data, we propose a detection system that prevents electric motorcycles from entering the elevator and deploys it to an edge-based Cambrian 1H8 edge-intelligent platform. We conduct privacy security modeling combining the application scenarios of the system proposed in this paper to improve the security of the system. In addition, to improve the system recognition accuracy, we fully analyze the challenges faced in the application scenarios and propose several data enhancement methods. This design can achieve low response time and high precision in real-time detection. Experimental results show that our system can achieve a high recall rate of 0.82. Moreover, by using data enhancement and data mixing strategies, it can reduce the misjudgment rate by 0.35 compared to using the original dataset. Simulated attack experiment shows that the *STI-1H8* model can recognise 100% of the application layer attacks, 81% of the network layer attacks, and 84% of the perception layer attacks. Comparative experiments show that compared with cloud computing, the edge computing solution reduced the latency by 19.6%. Moreover, compared with the mainstream electric motorcycle detection system *Kediou*, the power consumption of our system is only half that of the *Kediou* system, but its recall rate is improved by approximately 22%. Our proposed system has the advantages of higher security, higher accuracy, lower power consumption, and shorter detection intervals under multiple test indicators.

Abbreviations

AI: Artificial Intelligence; mAP: mean Average Precision; SSD: Single Shot MultiBox Detector; OSD: On-Screen Display; RTSP: Real Time Streaming Protocol; DRAM: Dynamic Random Access Memory; FPU: Floating-Point Unit; CNNs: Convolutional Neural Network; VGG: Visual Geometry Group; Wi-Fi: Wireless-Fidelity; YUV: Luminance-Bandwidth-Chrominance; VENC: Video Encoder; JSON: JavaScript Object Notation; SOC: System On Chip; S3FD: Single Shot Scale-invariant Face Detector; IoT: Internet of Things

Acknowledgements

This work was supported by the National Science Youth Fund of Jiangsu Province (no. BK20190224) and the Jiangsu Postdoctoral Science Foundation (no. 2019K224).

Authors' contributions

This paper is completed under the supervision of author Zongwei Zhu. Jing Cao wrote the paper. Tiancheng Hao is responsible for the technical architecture design, Wenjie Zhai is responsible for the experiment, and Bin Sun is responsible for the images. The grammar of the paper was reviewed and modified by Gangyong Jia. Finally, Ming Li gives some modification suggestions. All author(s) have read and approved the final manuscript.

Availability of data and materials

The 2000 raw images we use in the training dataset cannot be shared at this time as they are confidential. The VOC 2007 dataset is available at http://pjreddie.com/media/files/VOCtrainval_06-Nov-2007.tar.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Suzhou Research Institute, University of Science and Technology of China, Renai Road, Suzhou, China. [2]School of Information and Control Engineering, China University of Mining and Technology, University Road, Xuzhou, China. [3]School of Computer, Hangzhou Dianzi University, Wenyi Road, Hangzhou, China. [4]CCTEG Changzhou Research Institute, Changzhou, China.

Received: 24 February 2020 Accepted: 2 July 2020
Published online: 29 July 2020

References

1. Electric Vehicles Have Become a 'disaster Area' of Fire, with a Life-saving Index Beyond Imagination. https://my.mbd.baidu.com/ud8i9ii?f=cp&u=f2cd247418ce3b40. Accessed 15 Apr 2018
2. Circular of the Ministry of Public Security on Regulating the Parking and Charging of Electric Vehicles and Strengthening Fire Prevention. http://www.gov.cn/xinwen/2018-01/02/content_5252486.htm. Accessed 2 Jan 2018
3. Rakumthong W, Phetcharaladakun N, Wealveerakup W, Kamnoonwatana N (2014) Unattended and stolen object detection based on relocating of existing object. In: 2014 Third ICT International Student Project Conference (ICT-ISPC). IEEE, New York. pp 115–118
4. Xin H, Zeng D (2017) Real-time pedestrian warning system on highway using deep learning methods. In: 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS). IEEE, New York. pp 701–706
5. Gao H, Liu C, Li Y, Yang X (2020) V2VR: Reliable Hybrid-Network-Oriented V2V Data Transmission and Routing Considering RSUs and Connectivity Probability. pp 1–5
6. Prentice C, Karakonstantis G (2018) Smart office system with face detection at the edge. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, New York. pp 88–93
7. Gao H, Xu Y, Yin Y, Zhang W, Li R, Wang X (2020) Context-Aware QoS Prediction With Neural Collaborative Filtering for Internet-of-Things Services. IEEE Internet Things J 7(5):4532-4542
8. Ahmed A, Ahmed E (2016) A survey on mobile edge computing. In: 10th IEEE International Conference on Intelligent Systems and Control, (ISCO 2016). Vol. 70. pp 59-63
9. Yuan D, Zhu X, Mao Y, Zheng B, Wu T (2019) Privacy-preserving pedestrian detection for smart city with edge computing. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). pp 1–6. https://doi.org/10.1109/WCSP.2019.8927923
10. Barthélemy J, Verstaevel N, Forehead H, Perez P (2019) Edge-computing video analytics for real-time traffic monitoring in a smart city. Sensors 19(9):2048
11. Gao H, Huang W, Duan Y (2020) The Cloud-Edge Based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments: A QoS Prediction Perspective. ACM Trans Internet Technol
12. Ma X, Gao H, Xu H, Bian M (2019) An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing. EURASIP J Wirel Commun Netw 2019(1):249
13. Mambo M (2018) Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Gener Comput Syst 78(PT.2):680–698
14. Dai M (2013) Research on security threat modeling of internet of things perception layer. Inf Network Secur s1:9–12
15. Kruegel C, Mutz D, Robertson W, Valeur F (2003) Bayesian event classification for intrusion detection. In: 19th Annual Computer Security Applications Conference, 2003. Proceedings. IEEE, Los Alamitos. pp 14–23
16. Matzner S (1999) An application of machine learning to network intrusion detection. In: Computer Security Applications Conference. IEEE, Los Alamitos. pp 371–377
17. Cambricon 1H series product technology. http://www.cambricon.com/index.php?m=content&c=index&a=lists&catid=13. Accessed June 2020
18. Zhao G, Rong C, Jaatun M, Sandnes F (2012) Reference deployment models for eliminating user concerns on cloud security. J Supercomput 61(2):337–352

19. Wong JAH (1979) Algorithm AS 136: A K-Means Clustering Algorithm. J R Stat Soc 28(1):100-108
20. Gao H, Kuang L, Yin Y, Guo B, Dou K (2020) ?Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing Apps. Proc ACM/Springer Mobile Netw Appl (MONET)
21. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. arXiv:1409.1556v6
22. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRRabs/1704.04861. 1704.04861
23. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278-2324
24. Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu C, Berg AC (2015) Ssd: Single shot multibox detector. CoRRabs/1512.02325 9905. Springer, Cham. 1512.02325
25. Everingham M, Van Gool L, Williams C, Winn J, Zisserman A The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. Accessed May 2020
26. Salhaoui M, González A, Arioua M, Ortiz F, El Oualkadi A, Torregrosa C (2019) Smart industrial iot monitoring and control system based on uav and cloud computing applied to a concrete plant. Sensors 19:3316. https://doi.org/10.3390/s19153316
27. Xiao Y, Jia Y, Liu C, Cheng X, Yu J, Lv W (2019) Edge computing security: State of the art and challenges. Proc IEEE 107(8):1608–1631
28. Wei Y (2015) Research on security of iot perception layer based on node authentication. Netinfo Secur 11:27–32
29. Roukounaki A, Efremidis S, Soldatos J, Neises J, Walloschke T, Kefalakis N (2019) Scalable and configurable end-to-end collection and analysis of iot security data : Towards end-to-end security in iot systems. In: 2019 Global IoT Summit (GIoTS). IEEE, Piscataway. pp 1–6
30. Aborujilah A, Ismail M, Musa S (2014) Detecting tcp syn based flooding attacks by analyzing cpu and network resources performance. In: 2014 3rd International Conference on Advanced Computer Science Applications and Technologies. IEEE, New York. pp 157–161
31. Jiang L-s, Tian W-y, Zhu X-f (2011) Research on automatic detection of fake car-logo based on cloud-computing. In: 2011 International Conference on Multimedia Technology. IEEE, Piscataway. https://doi.org/10.1109/icmt.2011.6001967
32. Roman R, Lopez J, Mambo M (2018) Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Gener Comput Syst 78:680–698
33. Prentice C, Karakonstantis G (2018) Smart office system with face detection at the edge. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)IEEE. IEEE. pp 88–93. https://doi.org/10.1109/SmartWorld.2018.00050
34. Yuan D, Zhu X, Mao Y, Zheng B, Wu T (2019) Privacy-preserving pedestrian detection for smart city with edge computing. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, New York. https://doi.org/10.11092Fwcsp.2019.8927923
35. Barthélemy J, Verstaevel N, Forehead H, Perez P (2019) Edge-computing video analytics for real-time traffic monitoring in a smart city. Sensors 19(9):2048
36. Buda M, Maki A, Mazurowski M (2018) A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259
37. Ayan E, Ünver H (2018) Data augmentation importance for classification of skin lesions via deep learning. In: 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT). IEEE, Piscataway. pp 1–4
38. Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li S (2017) S3fd: Single shot scale-invariant face detector. In: Proceedings of the IEEE International Conference on Computer Vision. pp 192–201

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.