Zhou, Xiaokang; Xu, Xuesong; Liang, Wei; Zeng, Zhi; Yan, Zheng

Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT

Published in:
IEEE Internet of Things Journal

DOI:
10.1109/JIOT.2021.3077449

Published: 15/08/2021

Document Version
Peer reviewed version

Please cite the original version:
Zhou, X., Xu, X., Liang, W., Zeng, Z., & Yan, Z. (2021). Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT. *IEEE Internet of Things Journal*, *8*(16), 12588-12596. [9422817]. https://doi.org/10.1109/JIOT.2021.3077449

Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT

Xiaokang Zhou, *Member, IEEE,* Xuesong Xu, *Member, IEEE,* Wei Liang, *Member, IEEE,* Zhi Zeng, and Zheng Yan, *Senior Member*, *IEEE*

***Abstract*—Along with the rapid development of Cloud Computing, IoT, and AI technologies, cloud video surveillance (CVS) has become a hotly discussed topic, especially when facing the requirement of real-time analysis in smart applications. Object detection usually plays an important role for environment monitoring and activity tracking in surveillance system. The emerging edge-cloud computing paradigm provides us an opportunity to deal with the continuously generated huge amount of surveillance data in an on-site manner across IoT systems. However, the detection performance is still far away from satisfactions due to the complex surveilling environment. In this study, we focus on the multi-target detection for real-time surveillance in smart IoT systems. A newly designed deep neural network model called A-YONet, which is constructed by combining the advantages of YOLO and MTCNN, is proposed to be deployed in an end-edge-cloud surveillance system, in order to realize the lightweight training and feature learning with limited computing sources. An intelligent detection algorithm is then developed based on a pre-adjusting scheme of anchor box and a multi-level feature fusion mechanism. Experiments and evaluations using two datasets, including one public dataset and one homemade dataset obtained in a real surveillance system, demonstrate the effectiveness of our proposed method in enhancing training efficiency and detection precision, especially for multi-target detection in smart IoT application developments.**

***Index Terms*—Deep Learning, Neural Network, Object Detection, Edge Computing, `Smart IoT, Cloud Video Surveillance**

I. INTRODUCTION

Cloud video surveillance (CVS) has increasingly received widespread attention due to the high development of IoT and edge computing technologies. Recently, many smart applications are exploiting based on CVS systems, which focus on handling the continuously generated sequences of data from a whole surveillance environment. Accordingly, multi-object detection has become a significant technique and drawn lots of attentions from both industrial and academic fields [1-3]. Challenges include detecting and locating the moving objects, and then recognizing and extracting the instant features, which requires a strong real-time computing capability to efficiently handle the huge amount of IoT data in smart surveillance systems.

Conventional CVS system is usually implemented in a centralized computing manner. With the explosive growth of surveillance devices and continuous enhancement of large-scale of high-quality surveillance data in IoT environments, traditional paradigms with centralized processing are facing with more and more challenging issues. First, the gap between capabilities of data processing in cloud and speed of surveillance data generation is growing bigger. It is predicted that, only the data collected by the camera will exceed 869PB by the end of 2020. Second, the transmission of huge amounts of data across CVS systems requires high bandwidth, which calls for new techniques to realize the real time, low latency, energy efficiency, and high accuracy surveillance tasks [4]. It is essential to design a new edge-cloud surveillance infrastructure, in which the massive computing tasks are transferred from the central cloud server to edge servers. Thus, the well-trained AI models can be deployed in edge servers, to build a smart IoT system and effectively solve the above time-consuming problems in a distributed and reliable computing manner.

State-of-the-art techniques have achieved significant results for object detection in fixed static scenes, however, it still feels difficult when dealing with complex dynamic scenes. Tasks of object tracking and detection in dynamic scenes are often affected by multiple factors including location, density, moving state, and illumination changes. Moreover, current smart applications in dynamic surveilling environments, such as visual object detection [5], emotion analysis and identification recognition [6], require real-time data control, processing and communications [7]. Traditional strategies are no longer available in end devices, due to their high requirements on computing resources, management and storage capabilities. Therefore, it is necessary to exploit the lightweight learning model for intelligent object detection, and deploy the efficient

X. Zhou is with the Faculty of Data Science, Shiga University, Hikone, and RIKEN Center for Advanced Intelligence Project, Tokyo, Japan (e-mail: zhou@biwako.shiga-u.ac.jp).

X. Xu (corresponding author) is with the Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, Changsha, China, Corresponding author (e-mail: xuxs@hutb.edu.cn).

W. Liang (corresponding author) is with the Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, Changsha, China (e-mail: weiliang@csu.edu.cn).

Z. Zeng is with the Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, Changsha, China (e-mail: zhizeng416416@163.com).

Z. Yan is with the State Key Laboratory on Integrated Services Networks and the School of Cyber Engineering, Xidian University, Xi'an, China, and Department of Communications and Networking, Aalto University, Espoo, Finland (e-mail: zyan@xidian.edu.cn).

training process in edge devices [8, 9], in order to realize the newly established edge-cloud surveillance with smart IoT application development.

In this study, to overcome the above problems and meet the need of real-time control and analysis in big data surveilling environments, we aim to implement a distributed end-edge-cloud surveillance with smart IoT system. In particular, an intelligent object detection method is proposed to recognize multiple moving targets with different sizes in dynamic scenes. To optimize the limited computing sources, the surveillance data is compressed and key features are extracted in the end. A lightweight learning model with a newly designed deep neural network structure is deployed in the edge, which may efficiently improve the training process, and further facilitate the smart application development in the cloud. Our contributions are mainly concluded as follows.

i) A three-layer of "end-edge-cloud" architecture is built, which can realize data compression and feature extraction in the end, multi-level feature fusion and lightweight model training in the edge, and smart application deployments in the cloud, for the real-time surveillance and analysis in intelligent IoT systems.

ii) An integrated learning model, called A-YONet, is proposed, in which a newly designed network structure is constructed by combining the advantages of YOLO and Multi-Task Convolutional Neural Network (MTCNN), and a multi-level feature fusion mechanism is developed to enhance the training efficiency during feature mapping for multiple targets with different sizes.

iii) A multi-target detection algorithm is developed, in which a clustering-based pre-adjusting scheme for anchor box is designed to improve the precision when coping with multiple moving objects in real surveillance systems.

The rest of this article is organized as follows. Section II presents an overview of related works. In Section III, we introduce the infrastructure of the end-edge-cloud surveillance system, and the detailed network architecture of our proposed learning model. Algorithm and mechanism for multi-target detection are discussed in Section IV. In Section V, we demonstrate the experiment and evaluation results using real-world datasets. We conclude this study and give a promising perspective on future research in Section VI.

II. Related Work

A. Cloud Video Surveillance with Edge Computing

In IoT application environments, surveillance data is captured by a series of sensor devices distributed in various places. Considering its advantages of low bandwidth, low latency, high reliability and scalability, edge computing has been successfully applied to many fields, such as medical care [10], autonomous driving [11], remote acoustic detection [12], and achieved satisfactory results. Ananthanarayanan et al. [13] discussed the application of edge computing to CVS. They argued that this would make it possible to build a low-cost surveillance analysis system, and solve the problem of high demand for latency and bandwidth in surveillance applications. In particular, Xu and Helal [14] proposed an edge computing enabled architecture which considered the expansion of IoT with its scalability features. An event-driven programming model was built to deal with scaling behaviors and increase the flexibility of this newly designed architecture. To improve the transmission efficiency, Guo et al. [15] introduced an adaptive compression scheme, which could help end devices compress and transmit data without reducing the accuracy of target detection. Sun et al. [16] presented a video usefulness model based on edge computing, which could quickly screen out faulty devices for online failure detection, and thus relieved the network bandwidth across large-scale video surveillance systems. To reduce the workload of the backbone network and cloud, Wang et al. [17] adopted a resource allocation scheme which could dynamically adjust the computing resources among the three layers of end, edge, and cloud, so as to adapt to the actual situation.

B. Intelligent Object Detection in IoT

Intelligent object detection is an important technology in the field of computer vision, which plays a significant role in solving many problems related to IoT in modern society with video surveillance and remote control [18]. Extracting the valuable information from background area has been proved as effective means for object detection. Ebadi et al. [19] proposed an approximated robust principal component analysis method, to deal with the sequential surveillance data and obtain crisp object regions based on a dynamic tree-structured sparse matrix. Recently, deep learning models have been employed to improve object detections in IoT systems with smart sensors, which shows distinctive ability in pursuing higher accuracy and efficiency. Ren et al. [20] integrated a region proposal network with the Faster-RCNN, in which their shared convolutional features were utilized to improve the accuracy of real-time object detection. Peng et al. [21] designed a environment estimation scheme, in which the non-convex geometry information and texture information were incorporated into a general light reflection model, in order to improve the composite detection efficiency and optimize the lighting estimation accuracy. Liu et al. [22] constructed a single deep neural network, which could generate multiple feature maps to detect objects with various sizes. It is noted that traditional learning models may consume lots of computing resources, thus are not suitable to be directly deployed in IoT devices. Newly designed architectures with a small number of parameters, such as KPNet [23] and SquuezeNet [24], were considered to facilitate intelligent object detections in modern surveillance systems. Ahmed et al. [25] investigated applications of different CNN algorithms in IoT domain. In particular, they tested the Faster-RCNN and Mask-RCNN models trained by a new overhead view dataset. The results demonstrated the potential of these models in multiple object detections. Guo et al. [26] focused on context-aware object detection in an edge-cloud cooperation way. They built a deep learning based model in cloud server, and utilized a message-passing method to explore features based on spatial relations between objects and adjust the detection model.

III. Intelligent End-Edge-Cloud Surveillance System

In this section, the infrastructure of an intelligent end-edge-cloud surveillance system with its fundamental function modules is introduced. A framework of newly designed neural network model is then discussed to realize the multi-target detection based on a deep learning scheme in smart IoT systems.

A. Infrastructure for Intelligent End-Edge-Cloud Surveillance System

To tackle the real-time object detection and analysis of huge amount surveillance data across IoT networks in CVS environments, a three-layer "end-edge-cloud" surveillance system based on different IoT devices is designed. Precisely, it includes the End Data Processing Layer, Edge Model Training Layer, and Cloud Application Development Layer, to realize the data acquisition and compression in the end, multi-feature fusion and lightweight model training in the edge, and smart applications in the cloud respectively. The system infrastructure with detailed functional components is illustrated in Fig. 1.

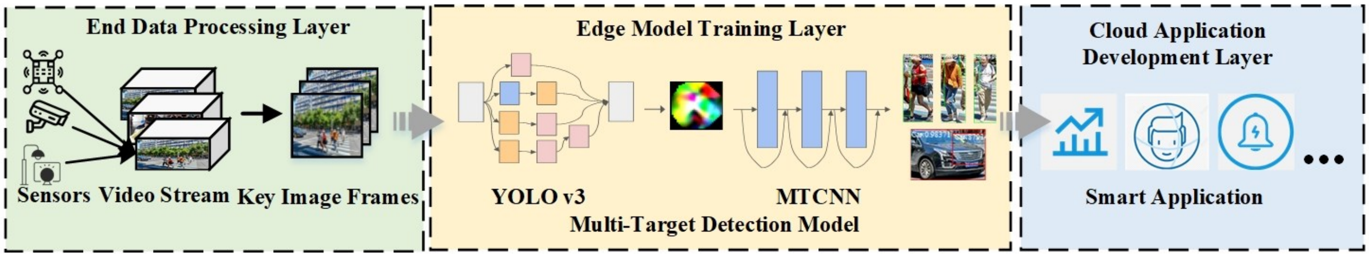

End Data Processing Layer
Sensors
Video Stream
Key Image Frames
Edge Model Training Layer
YOLO v3
MTCNN
Multi-Target Detection Model
Cloud Application Development Layer
Smart Application
•••

Fig.1. Infrastructure of End-Edge-Cloud Surveillance System in Smart IoT

End Data Processing (EDP) Layer: This layer is in charge of data collection and processing based on several sets of IoT sensors. Taking the video stream captured by cameras as an example, the data stream in this layer will be split into several segments, and only the key image frames containing the important information for target identification will be transmitted to the next layer.

Edge Model Training (EMT) Layer: This layer is mainly responsible for completing data analysis and target detection. We deploy a lightweight learning model using an improved deep neural network structure in this layer, which can bring in a better balance between the detection efficiency and resource consumption.

Cloud Application Development (CAD) Layer: This layer provides a programming platform and operating environment for different applications. Connecting with the cloud database, smart applications, such as event alarm and traffic statistics, can be safely deployed in this layer. In addition, this layer can collaborate with the EMT layer to balance the compute power as well.

Based on the three-layer "end-edge-cloud" system discussed above, the problem investigated in this study can be formalized as follows. Given a generated image frame set $F = \{f_1, f_2, \dots\}$, in which we assume each image frame f_i contains several target objects, the goal of this study is to design an efficient detection method to identify as many objects as possible in each f_i. To address this problem, G_i is defined to represent the set of actual objects for f_i as the ground truth, in which g_{ij} is the j_th element of G_i. More precisely, g_{ij} is composed of the detailed location of the object as $\left(x_{g_{ij}}, y_{g_{ij}}, w_{g_{ij}}, h_{g_{ij}}\right)$ and its corresponding class as $class_{g_{ij}}$. T_i is defined to represent the set of detected objects for f_i, in which t_{il} is the l_th element of T_i. Likewise, t_{il} is composed of $(x_{t_{il}}, y_{t_{il}}, w_{t_{il}}, h_{t_{il}})$ and the corresponding $class_{t_{il}}$ as well. Accordingly, the goal of this study can be formulated as to minimize the gap between the detected targets $(x_T, y_T, w_T, h_T, class_T)$ and the ground truth $(x_G, y_G, w_G, h_G, class_G)$ for all the image frames in F, which can be addressed as follows.

$$\Phi\text{=arg}\min_{T} \sum\nolimits_i \sqrt{\left(x_{T_i} - x_{G_i}\right)^2 + \left(y_{T_i} - y_{G_i}\right)^2}$$
$$\text{s.t.} \begin{cases} class_{G_i} = class_{T_i} \\ \left|w_{T_i} \times h_{T_i} - w_{G_i} \times h_{G_i}\right| \le \varepsilon \\ \left|w_{T_i} - w_{G_i}\right| + \left|h_{T_i} - h_{G_i}\right| \le \epsilon \end{cases} \tag{1}$$

where $0 < \varepsilon < 0.01$ and $0 < \epsilon < 0.1$ are the corresponding thresholds.

B. Basic Framework of A-YONet Model

We propose a multi-target detection model based on the combination of YOLOv3 and MTCNN, which is called as Advanced YONet (A-YONet), and deploy it in the "end-edge-cloud" surveillance system to improve the detection performance from the large amount of CVS data. Specifically, YOLOv3, an end-to-end method for object detection based on non-regional candidates, is employed and developed for feature extraction. Since the traditional YOLO model utilizes so many convolution layers, which results in a large storage usage and low detection speed in constrained environments, we introduce the O-Net, the last component in MTCNN, as the filter of candidate targets. The basic framework of the A-YONet is illustrated in Fig. 2.

The default size of the input data in the A-YONet is 416 x 416, which will be divided into $S \times S$ grids with the same width and height for feature extraction. The Darknet-53 without the fully connection layer, which is the backbone network of YOLOv3 and consists of a series of CBL (i.e., convolution layers, batch normalization layers, and Leaky ReLU layers) units, is selected as the feature extractor in the model. To get a smaller, faster and better model that can run in constrained environments, we use 1×1 convolution layer to replace the usually used 3×3 convolution layer when the size of filters is 1024. Furthermore, we design a feature fusion strategy to integrate the low-level and high-level feature maps for better feature representations. The size of the anchor box is re-adjusted based on a clustering scheme, which can accelerate the convergence compared with the traditional random selection scheme. Finally, the O-Net in MTCNN based on the shallow network, is utilized to filter the candidates, and pursue a high efficiency for multi-target detection.

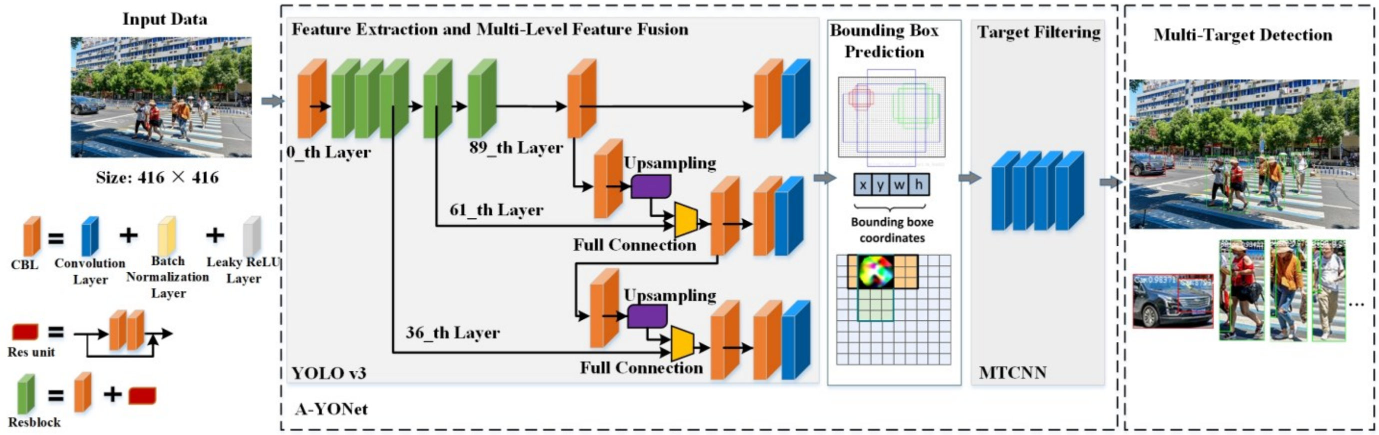


Fig.2. Network Architecture of A-YONet

IV. Multi-Target Oriented Object Detection in Smart IoT System

In this section, we discuss the detailed mechanism and implementation of the proposed A-YONet, including the anchor box adjusting mechanism, multi-level feature fusion strategy, and multi-target detection algorithm.

A. Pre-Adjusting Scheme for Anchor Box

We introduce a clustering scheme to adjust the anchor boxes in terms of the prediction of bounding boxes in the A-YONet model. Following the training of YOLO model, we can obtain $(x_\lambda, y_\lambda, w_\lambda, h_\lambda)$ to represent the translational positions and the corresponding changes of size for the anchor box, which can be used to predict four coordinates for each bounding box. Given the n_th bounding box b_{in} for target t_{il}, the prediction of its $(x_{b_{in}}, y_{b_{in}}, w_{b_{in}}, h_{b_{in}})$ is shown in Fig. 3, and can be formulated as follows.

$$\begin{aligned} x_{b_{in}} &= \sigma(x_\lambda) + x_{grid} \\ y_{b_{in}} &= \sigma(y_\lambda) + y_{grid} \\ w_{b_{in}} &= w_{a_m} \cdot e^{w_\lambda} \\ h_{b_{in}} &= h_{a_m} \cdot e^{h_\lambda} \end{aligned} \tag{2}$$

where w_{a_m} and h_{a_m} indicate the corresponding width and height of the m_th anchor box a_m. Function $\sigma(*)$ is utilized to normalize the value between 0 and 1. (x_{grid}, y_{grid}) represents the position of the upper left corner of the current grid.

A cluster center set C is defined to describe characteristics in terms of the size of objects, which will be further used to adjust the size of the anchor box. Concretely, we randomly select K elements from $\{G_i\}$ as the cluster center to initialize the set C, and the clustering will be generated based on the distance calculation according to the width and height $\left(w_{g_{ij}}, h_{g_{ij}}\right)$ of each element g_{ij}. Following this process, we can assign all the elements in $\{G_i\}$ to its closest center, and generate K clusters. We keep update the cluster center C by calculating the average width and height until C does not change anymore.

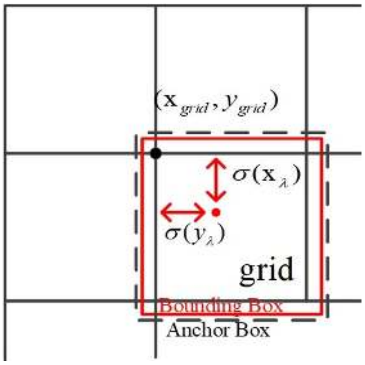


Fig.3. Prediction of Position for Bounding Box

Given c_o represents the o_th cluster center of C, the distance $d(g_{ij}, c_o)$ between g_{ij} and c_o can be calculated as follows.

$$d(g_{ij}, c_o) = 1 - IoU(g_{ij}, c_o) \tag{3}$$

where $IoU(g_{ij}, c_o)$ indicates the ratio of the overlap part of the two regions g_{ij} and c_o. The detailed calculation can be expressed in Eq. (4).

$$IoU(g_{ij}, c_o) = \frac{S_{g_{ij} \cap c_o}}{S_{g_{ij} \cup c_o}} \tag{4}$$

where $S_{g_{ij} \cap c_o}$ is the intersection of g_{ij} and c_o, while $S_{g_{ij} \cup c_o}$ is the corresponding union of g_{ij} and c_o.

B. Multi-Level Feature Fusion Mechanism

The Darknet-53 network in YOLOv3 has excellent feature extraction performance, because it calculates the feature hierarchy layer by layer based on a series of convolution networks. Feature maps can then be generated based on the feature hierarchy with different spatial resolutions by increasing the step size of convolution kernel. Consequently, it gradually reduces the $N \times N$ input key image frame to $\frac{1}{32} \times N \times N$. The generation process of each layer of feature map is described as follows.

$$fm_{ir} = \begin{cases} Action(f_i), & r = 0 \\ Action(fm_{ir-1}), & r = 1,2,\ldots,74 \end{cases} \tag{5}$$

where fm_{ir} indicates the r_th layer feature map of f_i, $Action \in \{act_{Conv}, act_{BN}, act_{LR}, act_{Add}\}$ stands for the action of convolution, batch normalization, Leaky ReLU activation function, and tensor addition, in the r_th layer.

We thus introduce a multi-level feature fusion strategy to realize the different scales of fusion of feature maps. Specifically, the shape of the feature hierarchy is set as pyramid, and the low-resolution features with strong semantics and high-resolution features with weak semantics are fused together based on deep-shallow connections. Thus, the detection accuracy can be improved by making a full use of all feature maps. The detailed fusion process can be described as follows.

$$ffm_{ip} = act_{Conv}(act_{FC}(act_{US}(fm_{ir_{high}}), fm_{ir_{low}}))) \tag{6}$$

where ffm_{ip} indicates the p_th fused feature map. $fm_{ir_{high}}$ and $fm_{ir_{low}}$ indicate the high-resolution feature map and low-resolution feature map respectively. act_{US} stands for the action of upsampling, act_{FC} stands for the action of full connection, and act_{Conv} stands for the action of convolution.

During this feature fusion process, the size of the corresponding feature map $fm_{ir_{high}}$ will be expanded by 2*2 times while the number of channels will remain the same after upsampling, which will then be connected with the closest feature map $fm_{ir_{low}}$ of the same size after $act_{US}(fm_{ir_{high}})$. For instance, when $p = 1, 2, 3$, after the final convolution, the size of fused feature map $ffm_{i1}, ffm_{i2}, ffm_{i3}$ will become $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$ of the original input sample respectively.

Accordingly, in each fused feature map, the appropriate anchor box will be chosen based on the proposed clustering scheme to generate bounding boxes, and the O-Net in MTCNN can then filter the candidate targets with the corresponding bounding boxes more quickly based on the multi-level feature fusion.

C. Multi-Target Detection Algorithm

We design a multi-target detection mechanism based on the A-YONet model, to facilitate the intelligent detection in CVS, which can be described in Algorithm 1.

According to Algorithm 1, the main structure of YOLO model ensures the fast speed in object detection, and the fusion of multi-level features can efficiently improve the accuracy in multi-target detection. In addition, the adjusted anchor box according to characteristics in terms of the size of objects can help to generate a bounding box which is more reasonable in an actual situation. Finally, the integration of MTCNN can enhance the detection precision by timely filtering the candidate targets for real-time analysis of moving objects in CVS.

Algorithm 1: Multi-target detection algorithm based on A-YONet

```
Input: The key image frameset retrieved from video stream $F$
       The ground truth $\{G_i\}$
Output: The trained model $M_{\text{A-YONet}}$

Initialize the loss threshold $\sigma$ for $M_{\text{A-YONet}}$
Initialize $loss = \infty$
Initialize $K = 3 * q, q \in \{1,2,...\}$
Randomly select $K$ elements from $\{G_i\}$ to initialize the set $C$
while $C$ changed do:
    for each $G_i$ do:
        for each $c_o$ in $C$ do:
            Calculate the distance $d(g_{ij}, c_o)$ for each element $g_{ij}$
            in $G_i$ by Eq. (3) and (4)
        end for
        Assign $g_{ij}$ to the closest center
    end for
    Update $C$
end while
$\{a_m\} = C$
while $loss > \sigma$ do:
    for each $f_i$ from $F$ :
        $FM = \emptyset$
        Get feature map $fm_{ir}$ by Eq. (5) for each layer $r$ in
        Darknet-53 and set $FM = FM \cup fm_{ir}$
        for each $fm_{ir_{high}}$ :
            Fuse the lower- resolution feature maps $FM$ and
            current (high- resolution) feature maps $fm_{ir_{high}}$ to
            generate the fused feature maps $ffm_{ip}$ by Eq. (6)
        end for
        Obtain the bounding box set $B_i = \{b_{in}\}$ for $f_i$ based
        on $ffm_{ip}$ by Eq. (2) with $\{a_m\}$
        Obtain $T_i$ by filtering bounding boxes $B_i$ with O-Net
    end for
    Update $loss$ according to the prediction error between
    $\{T_i\}$ against $\{G_i\}$
end while
return $M_{\text{A-YONet}}$
```

V. Experiment and Analysis

In this section, to investigate the effectiveness of the proposed A-YONet model for multi-target detection, we conduct evaluation experiments based on two datasets, including one public dataset and one homemade dataset obtained from the CVS system respectively.

A. Dataset and Experiment Design

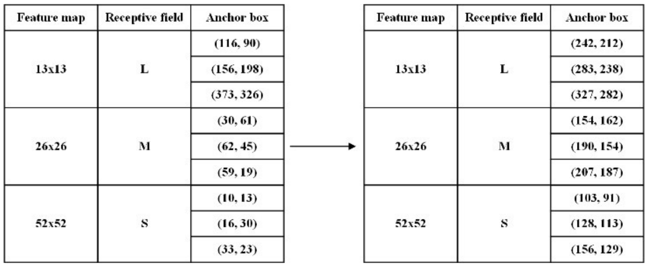
To train and test our model in resource-constrained embedded platform, we choose Jetson TX1 with 6GB memory and 256 CUDA cores NVIDIA Maxwell GPU as the single-board chip, which has a computing power of one TFLOPs and a peak power consumption of only 10W. We set gn6i as our server, which has 2560 CUDA cores and a computing power of 65 TFLOPs. The training parameters of the A-YONet are shown in Table I.

The PASCAL VOC dataset is widely used for evaluation of classification and detection, which contains 20 kinds of objects ranging from person, animals, vehicles and household. Unlike images of specific light and background taken in the laboratory, PASCAL VOC consists of images collected from the Internet, and each image includes several kinds of objects. Thus, it can be used to evaluate algorithms in different scenes and be more similar to the real detection environment for surveillance. Specifically, the VOC 2007 and 2012 are chosen as the training and validation datasets in this experiment, which contain 16551 images and 40058 objects with annotations of classes and positions.

TABLE I
BASIC SETTING OF NETWORK PARAMETERS

Parameter	Setting Value
Learning Rate	0.0001
Epoch	5000
Batch Size	32
Weight Decay	0.0005

In addition to the public dataset, we deploy the proposed intelligent surveillance system in the university campus, which results in a large homemade dataset based on several real scenes, including 7043 images from classrooms and other public areas. Objects covered in this dataset include person, books, bags, bottles, cell phones, and etc. The re-adjusted sizes of anchor box based on our clustering scheme are shown in Fig. 4.



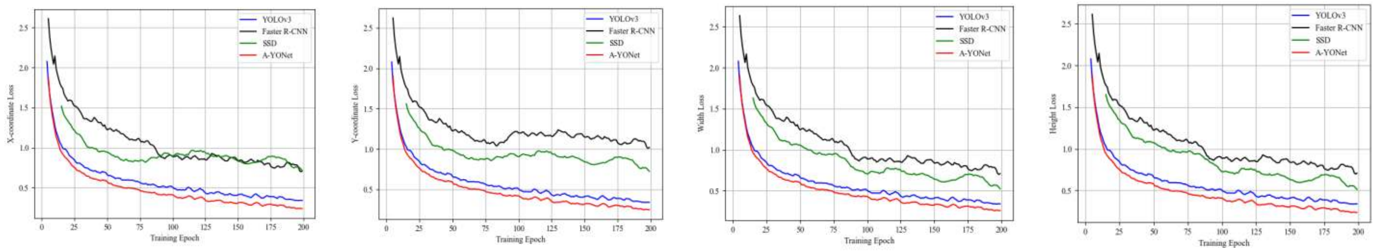
Feature map	Receptive field	Anchor box
13x13	L	(116, 90)
		(156, 198)
		(373, 326)
26x26	M	(30, 61)
		(62, 45)
		(59, 19)
52x52	S	(10, 13)
		(16, 30)
		(33, 23)

Feature map	Receptive field	Anchor box
13x13	L	(242, 212)
		(283, 238)
		(327, 282)
26x26	M	(154, 162)
		(190, 154)
		(207, 187)
52x52	S	(103, 91)
		(128, 113)
		(156, 129)

Fig.4. Adjusted Size of Anchor Box

B. Comparison Experiment Using Public Dataset

We first compare the training efficiency of our method with three traditional object detection methods: YOLOv3, Faster R-CNN, and SSD. As shown in Fig. 5, it is found that A-YONet can better locate the grid of candidate objects and converge faster during the training process. Because the proposed method uses a more precise mechanism for anchor box adjusting, which can efficiently reduce the losses in height and width of the bounding box, and accurately locate the coordinate of those candidates.



(a) (b) (c) (d)

Fig.5. Training Efficiency Comparisons Among Different Methods

Table II shows the comparison results among the four methods according to Average Intersection-over-Union (IoU) and Frames Per Second (FPS). Average IoU refers to the average value of the coincidence degree between the detected candidate bounding box and the ground truth bounding box. The larger the Average IoU, the more consistent the detected result is with the actual situation. FPS represents the number of frames detected per second, which can be used to describe the speed of the object detection.

TABLE II
PERFORMANCE COMPARISONS FOR DIFFERENT DETECTION METHODS.

Method	YOLOv3	Faster R-CNN	SSD	A-YONet
Average IoU	0.756	0.774	0.762	**0.787**
FPS	**49**	15	17	48

As shown in Table II, according to Average IoU, the performance of A-YONet is 0.031 higher than YOLOv3, 0.025 higher than SSD, and 0.13 higher than Faster R-CNN. This can be explained as the adjustment of the anchor box can efficiently improve the detection performance. On the other hand, according to FPS, the A-YONet is much faster than Faster R-CNN and SSD, and only a little slower than YOLOv3. This result indicates that the combination of O-Net will not lose a lot of time advantage.

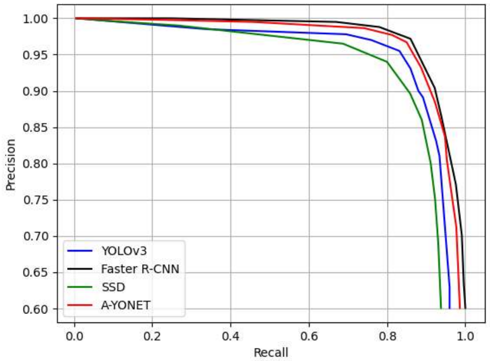


Fig.6. Comparisons Based on Precision-Recall Curve for Different Methods

Furthermore, Fig. 6 shows the results based on Precision-Recall curve for all the four methods. The results demonstrate that the performance of A-YONet is between Fast R-CNN and YOLOv3. In addition, we evaluate detection performances of the four methods among different classes according to Mean Average Precision (mAP). As shown in Table III, the A-YONet improves the detection precision by 5.4% compared with YOLOv3, 2.0% compared with Faster R-CNN, and nearly 9.6%

compared with SSD. In details, performances of the A-YONet in all the 20 kinds of objects are better than YOLOv3 and SSD. Except classes of bottle, potted plant, and sheep, performances of the A-YONet in the other 17 kinds of objects are better than Faster R-CNN.

TABLE III
COMPARISONS AMONG DIFFERENT CLASSES

Class	YOLOv3 (%)	Faster R-CNN (%)	SSD (%)	A-YONet (%)
Aero plane	84.5	85.4	83.1	**86.1**
Bicycle	80.2	86.2	81.3	**87.8**
Bird	71.6	73.6	73.9	**74.2**
Boat	68.7	69.2	58.9	**69.9**
Bottle	67.1	**68.9**	51.2	67.4
Bus	84.3	87.9	79.8	**88.3**
Car	75.7	84.5	75.8	**85.1**
Cat	84.1	88.9	88.3	**89.2**
Chair	64.7	66.2	51.9	**67.1**
Cow	77.6	82.7	78.1	**83.5**
Dining table	56.7	74.4	58.2	**75.7**
Dog	84.3	86.7	87.6	**89.0**
Horse	82.4	82.1	82.2	**84.3**
Motorbike	82.5	85.7	80.9	**83.9**
Person	79.8	81.7	81.4	**83.2**
Potted plant	57.6	**60.7**	40.6	59.3
Sheep	76.8	**78.6**	72.7	77.4
Sofa	61.7	78.8	62.5	**79.6**
Train	83.8	82.3	83.2	**84.2**
Tv/monitor	62.1	78.7	68.9	**79.2**
mAP	74.3	77.7	70.1	**79.7**

C. Performance Evaluation in CVS System

We go further to compare detection performances of the four methods in a typical CVS system. Fig. 7 shows the precision results of different methods in detecting five kinds of objects in real-world. Obviously, the A-YONet achieves a better performance than the other three methods. This is because the proposed method efficiently leverages the fused features in both high and low layers, which is very important in improving the detection precision.

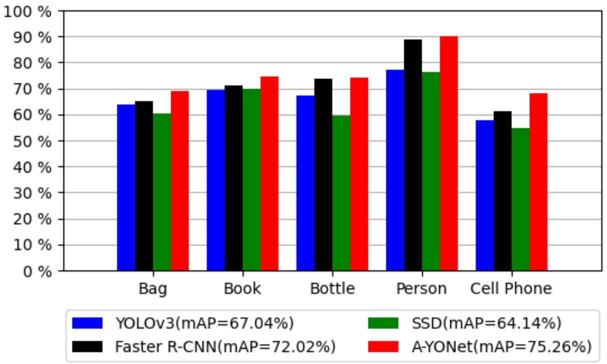
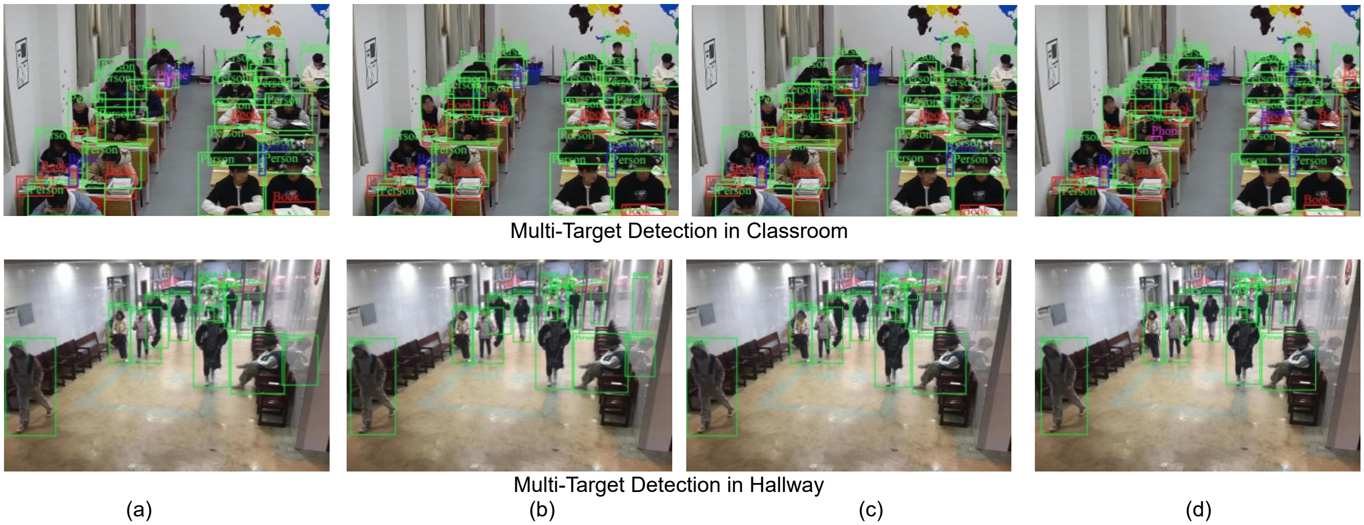


Fig.7. Precision Results of Different Methods on Five Different Objects

Fig. 8 demonstrates detection results of the four methods in two different surveilling scenarios, namely the classroom and hallway. It can be observed that performances of YOLOv3 and SSD are worse than Faster R-CNN. Although the Faster R-CNN can detect almost all the people in the two scenarios, some targets are wrongly detected and a few of books and bottles are missed. Contrastively, the A-YONet model outperforms the other three methods in both classroom and hallway environments, which indicates the outstanding capability of our method in multi-target detection among different real surveilling environments.


Multi-Target Detection in Classroom

Multi-Target Detection in Hallway

(a) (b) (c) (d)

Fig.8. Performance Comparisons on Multi-Target Detection. (a) YOLOv3. (b) Faster R-CNN. (c) SSD. (d) A-YONet

VI. CONCLUSION

In this paper, we investigated the real-time surveillance in smart IoT systems, and proposed a multi-target detection method, which could be deployed in an end-edge-cloud surveillance system, to facilitate the lightweight training and multi-level feature learning with limited computing sources in IoT environments.

In particular, a three-tier system infrastructure was introduced for the intelligent surveillance based on an end-edge-cloud architecture, in which the EDP layer was designed to deal with data compression and feature extraction in the end, the EMT layer was responsible for multi-level feature fusion and lightweight model training in the edge, and the CAD layer was employed for smart applications in the cloud. An integrated deep neural network model called A-YONet, which was constructed by utilizing both advantages of YOLO and MTCNN, was newly designed to realize the lightweight training and feature learning with limited computing sources. A pre-adjusting mechanism for anchor box was devised based on a clustering scheme to dynamically change the bounding boxes, and a multi-level feature fusion mechanism was introduced to enhance the training efficiency when handling multiple targets with different sizes during feature mapping processes. A multi-target detection algorithm was finally developed, which could improve the precision when dealing with multiple moving objects for real-time surveillance. Experiments were designed and conducted using two datasets: one public dataset and one homemade dataset in a real surveillance system. Evaluation results demonstrated the effectiveness of our proposed model and method for real-time surveillance in smart IoT systems, comparing with three baseline methods.

In the future, we will further study more deep learning schemes to enhance the detection accuracy and efficiency in the edge computing environment. More evaluations in different dynamic scenes will be investigated to improve the adaptability of our method.

Acknowledgment

The work was supported in part by the National Key R&D Program of China No. 2017YFE0117500, No. 2019YFE0190500, No.2020YFC0832700, No.2019YFB1705200, and No. 2019GK1010, in part by the National Natural Science Foundation of China No. 72088101, No. 91846301, No. 72091515, No.71991463 and No. 62072171, and in part by the Natural Science Foundation of Hunan Province of China No. 2019JJ40150.

References

[1] H. F. Nweke, Y. W. Teh, G. Mujtaba and M. A. Al-garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," Information Fusion, vol. 46, pp. 147-170, 2019.

[2] S. Zafeiriou, C. Zhang and Z. Zhang, "A survey on face detection in the wild: Past, present and future," Computer vision and image understanding, vol. 138, pp. 1-24, 2015.

[3] P. Albano, A. Bruno, B. Carpentieri, A. Castiglione, A. Castiglione, F. Palmieri, R. Pizzolante, K. Yim and I. You, "Secure and distributed video surveillance via portable devices," Journal of Ambient Intelligence and Humanized Computing, vol. 5, pp. 205–213, 2014.

[4] X. Zhang, Y. Cao, L. Peng, J. Li, N. Ahmad and S. Yu, "Mobile Charging as a Service: A Reservation-Based Approach," IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1976-1988, 2020.

[5] H. Zhang, K. F. Wang and F. Y. Wang, "Advances and Perspectives on Applications of Deep Learning in Visual Object Detection," Acta Automatica Sinica, vol. 43, no. 8, pp. 1289-1305, 2017.

[6] G. Antipov, M. Baccouche and J. Dugelay, "Face aging with conditional generative adversarial networks," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, pp. 2089-2093, 2017.

[7] Y. Cao, X. Zhang, B. Zhou, X. Duan, D. Tian and X. Dai, "MEC Intelligence Driven Electro-Mobility Management for Battery Switch Service," IEEE Transactions on Intelligent Transportation Systems, 2020. doi: 10.1109/TITS.2020.3004117

[8] J. Wang, J. Hu, G. Min, A. Y. Zomaya and N. Georgalas, "Fast Adaptive Task Offloading in Edge Computing Based on Meta Reinforcement Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 1, pp. 242-253, 2021.

[9] Z. Chen, J. Hu, G. Min, A. Y. Zomaya and T. El-Ghazawi, "Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 4, pp. 923-934, 2020.

[10] G. Muhammad, M. F. Alhamid, M. Alsulaiman and B. Gupta, "Edge Computing with Cloud for Voice Disorder Assessment and Treatment," in IEEE Communications Magazine, vol. 56, no. 4, pp. 60-65, April 2018.

[11] A. Meslin, N. Rodriguez and M. Endler, "Scalable Mobile Sensing for Smart Cities: The MUSANet Experience," IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5202-5209, June 2020, doi: 10.1109/JIOT.2020.2977298.

[12] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian and V. C. M. Leung, "Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring," IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 1, pp. 64-74, January 2019.

[13] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath and S. Sinha, "Real-Time Video Analytics: The Killer App for Edge Computing," Computer, vol. 50, no. 10, pp. 58-67, 2017.

[14] Y. Xu and A. Helal, "Scalable Cloud–Sensor Architecture for the Internet of Things," IEEE Internet of Things Journal, vol. 3, no. 3, pp. 285-298, June 2016, doi: 10.1109/JIOT.2015.2455555.

[15] Y. Guo, B. Zou, J. Ren, Q. Liu, D. Zhang and Y. Zhang, "Distributed and Efficient Object Detection via Interactions Among Devices, Edge, and Cloud," IEEE Transactions on Multimedia, vol. 21, no. 11, pp. 2903-2915, Nov. 2019.

[16] H. Sun, W. Shi, X. Liang and Y. Yu, "VU: Edge Computing-Enabled Video Usefulness Detection and its Application in Large-Scale Video Surveillance Systems," IEEE Internet of Things Journal, vol. 7, no. 2, pp. 800-817, Feb. 2020.

[17] J. Wang, J. Pan, F. Esposito, "Elastic urban video surveillance system using edge computing," Proceedings of the Workshop on Smart Internet of Things, 2017.

[18] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke and Z. Xiong, "Multiplex Labeling Graph for Near-Online Tracking in Crowded Scenes," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 7892-7902, 2020.

[19] S. E. Ebadi and E. Izquierdo, "Foreground Segmentation with Tree-Structured Sparse RPCA," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 9, pp. 2273-2280, 2018.

[20] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.

[21] B. Peng, W. Wang, J. Dong and T. Tan, "Optimized 3D Lighting Environment Estimation for Image Forgery Detection," IEEE Transactions on Information Forensics and Security, vol. 12, no. 2, pp. 479-494, 2017.

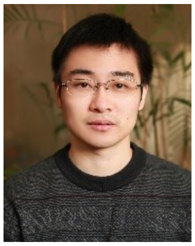
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, "Ssd: single shot multibox detector," Computer Vision - ECCV 2016, Lecture Notes in Computer Science, vol. 9905, pp. 21-37, 2016.

[23] G. Song, Y. Liu, Y. Zang, X. Wang, B. Leng, Q. Yuan, "Kpnet: towards minimal face detector", 34th AAAI Conference on Artificial Intelligence, vol. 34, no. 7, 2020.

[24] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-Excitation Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011-2023, 2020.

[25] I. Ahmed, S. Din, G. Jeon and F. Piccialli, "Exploring Deep Learning Models for Overhead View Multiple Object Detection," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5737-5744, 2020.

[26] J. Guo, B. Song, S. Chen, F. R. Yu, X. Du and M. Guizani, "Context-Aware Object Detection for Vehicular Networks Based on Edge-Cloud Cooperation," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5783-5791, 2020.



Xiaokang Zhou (M'12) is currently an associate professor with the Faculty of Data Science, Shiga University, Japan. He received the Ph.D. degree in human sciences from Waseda University, Japan, in 2014. From 2012 to 2015, he was a research associate with the Faculty of Human Sciences, Waseda University, Japan. He also works as a visiting researcher in the RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Japan, since 2017. Dr. Zhou has been engaged in interdisciplinary research works in the fields of computer science and engineering, information systems, and social and human informatics. His recent research interests include ubiquitous computing, big data, machine learning, behavior and cognitive informatics, cyber-physical-social systems, cyber intelligence and security. Dr. Zhou is a member of the IEEE CS, and ACM, USA, IPSJ, and JSAI, Japan, and CCF, China.

Xuesong Xu (M'12) received his M.S and Ph.D. degree in Control Science and Engineering from Hunan University in 2004 and 2009 respectively. From 2012 to 2015, he works as a post-doctoral at National University of defense and technology. From 2016-2017, he also works as a visiting researcher in Volen National Center for Complex Systems, Brandeis University, USA. Currently, He is a professor of Hunan University of Technology and Business, working at Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management. His research interests are in the areas of complex system optimization, Blockchain and Machine learning. He has published more than 30 papers at various conferences and journals, including FGCS, Energy reports, and Sensors. Prof. Xu is a member of the IEEE Computational Intelligence.

Wei Liang (M'19) received his M.S. and Ph.D. degrees in Computer Science from Central South University in 2005 and 2016. From 2005 to 2012, he worked in Microsoft (China) for soft engineering. From 2014 to 2015, he worked as an exchange researcher in the Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Japan. He is currently working at Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, China. His research interests include information retrieval, data mining, and artificial intelligence. He has published more than 20 papers at various conferences and journals, including FGCS, JCSS, and PUC. Dr. Liang is a member of the IEEE CS.

Zhi Zeng received her B.S. degree in Computer Science from Hunan University of Technology and Business. She is currently a graduate student in Management Science and Engineering, Hunan University of Technology and Business, China. Her major research interests is Artificial Intelligence and Blockchain.

Zheng Yan (M'06, SM'14) received the BEng degree in electrical engineering and the MEng degree in computer science and engineering from the Xi'an Jiaotong University, Xi'an, China in 1994 and 1997, respectively, the second MEng degree in information security from the National University of Singapore, Singapore in 2000, and the licentiate of science and the doctor of science in technology in electrical engineering from Helsinki University of Technology, Helsinki, Finland. She is currently a professor at the Xidian University, Xi'an, China and a visiting professor at the Aalto University, Espoo, Finland. Her research interests are in trust, security, privacy, and security-related data analytics. Prof. Yan serves as a general or program chair for 30+ international conferences and workshops. She is a steering committee co-chair of IEEE Blockchain international conference. She is also an associate editor of many reputable journals, e.g., IEEE Internet of Things Journal, Information Sciences, Information Fusion, JNCA, IEEE Access, SCN, etc.