

Received 5 August 2022, accepted 28 August 2022, date of publication 31 August 2022, date of current version 12 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203053

## RESEARCH ARTICLE

# Distributed Real-Time Object Detection Based on Edge-Cloud Collaboration for Smart Video Surveillance Applications

YUNG-YAO CHEN<sup>1</sup>, (Member, IEEE), YU-HSIU LIN<sup>2</sup>,  
YU-CHEN HU<sup>3</sup>, (Senior Member, IEEE), CHIH-HSIEN HSIA<sup>4</sup>, (Member, IEEE),  
YI-AN LIAN<sup>1</sup>, AND SIN-YE JHONG<sup>5</sup>

<sup>1</sup>Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan

<sup>2</sup>Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei 106344, Taiwan

<sup>3</sup>Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan

<sup>4</sup>Department of Computer Science and Information Engineering, National Ilan University, Ilan 260007, Taiwan

<sup>5</sup>Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan

Corresponding author: Yu-Hsiu Lin (yhlin@ntut.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 111-2221-E-027-050-, Grant MOST 111-3116-F-006-005-, and Grant MOST 111-3116-F-027-001-.

**ABSTRACT** Internet of Things (IoT) and artificial intelligence (AI) can realize the concept of “smart city.” Video surveillance in smart cities is, usually, based on a centralized framework in which large amounts of real-time media data are transmitted to and processed in the cloud. However, the cloud relies on network connectivity of the Internet that is sometimes limited or unavailable; thus, the centralized framework is not sufficient for real-time processing of media data needed for smart video surveillance. To tackle this problem, edge computing - a technique for accelerating the development of AIoT (AI across IoT) in smart cities - can be conducted. In this paper, a distributed real-time object detection framework based on edge-cloud collaboration for smart video surveillance is proposed. When collaborating with the cloud, edge computing can serve as converged computing through which media data from distributed edge devices of the network are consolidated by AI in the cloud. After AI discovers global knowledge in the cloud, it to be shared at the edge is deployed remotely on distributed edge devices for real-time smart video surveillance. First, the proposed framework and its preliminary implementation are described. Then, the performance evaluation is provided regarding potential benefits, real-time responsiveness and low-throughput media data transmission.

**INDEX TERMS** Cloud computing, edge computing, edge-cloud collaboration, object detection, video surveillance.

## I. INTRODUCTION

Recent breakthrough technologies are trending up for today's technologically driven society, i.e., from fundamental constituents of a city (e.g., smart homes, buildings and factories) to smart cities that promote our daily living by providing smart surveillance, healthcare, intelligent transportation and so on [1]. The technical combination of advances in Internet of Things (IoT) collecting data around a city [2] and

artificial intelligence (AI) using a wide range of algorithms in soft computing, machine learning, deep learning, image processing and computer vision to analyze collected data has been developed; this brings novel insights to turn the concept of “smart city” from hype to reality. Being the prominent technologies nowadays, IoT and cloud computing (i.e., the cloud) have received significant attention from both academia and industry, as they are used in numerous applications in smart homes including healthcare devices, manufacturing and cities [3]. Video surveillance is one of the main and important building blocks of smart cities, which offers people more

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen<sup>1</sup>.

tools and applications to monitor environments (e.g., pedestrians for security surveillance and management) and serves as a security force for improving a city's stability [2], [4]. Smart cities providing intelligent urban environments that can deliver a high quality of life to citizens to co-work with local government authorities to launch initiatives to be designed and implemented with smart technologies use a variety of software, user interfaces and communication networks alongside the AIoT (AI across IoT) to deliver connected solutions for the public. In the past decade, for video surveillance applications in different fields ranging from traffic monitoring to national security, it was a significant aspect to transmit directly huge amounts of real-time media data from end devices to the cloud via the Internet (or dedicated high-speed fiber networks) [5], [6]. The cloud with rich storage and computing resources can then perform massive-scale and complex data processing, to detect and identify valuable insights from vast amounts of received media data for surveillance applications. In such a centralized cloud-centered framework, this leads to a high investment (including annual maintenance costs) in network deployment (simultaneously transmitting IoT data to the cloud requires a large number of bandwidth resources, where the existing network infrastructure cannot support enough bandwidth at a reasonable price [7]). Also, it poses great challenges to the cloud in terms of high network latency and congestion from heavy communication burden [5], [7], [8]. Such a centralized cloud-centered framework has been gradually changed in recent years, as the current breakthrough technologies are driving us into the era of AIoT (AI across IoT). Limited by the network connectivity where the Internet is not always available, transmitting all media data from end devices to the cloud for data processing is no longer a wise approach for real-time video surveillance applications (latency-sensitive video surveillance applications) [9], [10]. Edge computing, a developed complement of cloud computing, can perform real-time media data processing at the edge of the Internet, while relieving computing and storage pressure of the cloud; this results in remedying high network latency and congestion [6], [8], [11], [12]. By leveraging storage and computing resources at the edge of the Internet, edge computing can provide distributed, real-time media data processing and analysis for surveillance applications [5], [8], [13]. Meanwhile, collaborating with the cloud to form edge-cloud computing, edge computing can also serve as converged computing through which data from distributed edge devices are consolidated through AI. After AI achieves global knowledge discovering, it is then deployed remotely as global knowledge sharing on distributed edge devices for real-time smart video surveillance [14]. Where, AI that is deployed at the edge of the network is trained in the cloud. The aforementioned advantages have promoted the significance of the edge collaborating with the cloud for developing next-generation video surveillance applications, where the cloud consolidates data from and interacts with edge devices to update their knowledge (trained local AI model).

Smart video surveillance is a new initiative that sets a higher ceiling than that of traditional video surveillance for the future of smart cities [3]. Yet, not much attention has been paid to smart video surveillance applications based on edge-cloud collaboration for smart cities in traffic monitoring, national security, healthcare and many others. In [15], an edge computing-based surveillance framework, a Convolutional Neural Network (CNN)-implemented edge device based on Raspberry Pi, for real-time human activity recognition is presented. The authors of [16] develop a face recognition system based on a fog computing platform, where a smartphone's photos are proceeded at the edge instead of the cloud, to achieve fast response of face recognition. The authors of [17] and [18] propose a framework for video stream acquisition, storage and analysis using a cloud based GPU (Graphic Processing Unit) cluster accelerating the video processing process to reduce the computational complexity. However, the studies only considered a centralized video analytics framework. In [19], [20], and [21], cloud-based smart video surveillance systems are shown, which were a centralized smart video surveillance paradigm. In [22], an edge computing-based video analytics for public safety is presented, which can distribute computing workloads in both the edge, including a smartphone, Raspberry Pi and body-worn cameras, and the cloud in an optimized way. An edge-cloud cooperation framework is proposed in [23], which aims at designing and implementing the framework in a heterogeneous converged communication network for edge computing in a coal mine environment. The authors of [24] investigate an edge-cloud computing based smart parking surveillance system for parking occupancy detection. Where, two detection methods are implemented at the edge and their detection results can be combined, for occlusion or extreme lighting conditions occurring for parking occupancy detection in a parking lot scene, by a rule or a metric on the cloud-centric server side for enhanced performance in parking occupancy detection. In [25], a cloud-based object tracking and behavior identification architecture performing abnormal fall detection based on a deep learning approach, a CNN, for a smart healthcare video surveillance application is proposed. As demonstrated in the study, the architecture purely depends on a cloud-based platform for object detection and fall recognition which is restricted to network bandwidth utilization. Where, the limitation can be easily resolved as a more robust process can be enforced in the architecture, utilizing the edge computing capability, within limited network bandwidth utilization. As reviewed above, it is vital to develop a dedicated edge-cloud computing architecture, utilizing an edge-cloud collaboration mechanism, for object detection and classification in smart video surveillance (a video-/image-processing related practical application) such that the full use of storage and computing resources of both the edge and the cloud to be leveraged within limited network bandwidth utilization and be enabled for autonomous AI updates so the whole system can be advanced can be realized. Therefore, in this paper, a distributed real-time object

detection and classification framework based on edge-cloud collaboration for smart video surveillance applications is proposed, and its preliminary implementation is demonstrated. Where, both the edge and the cloud can be leveraged within limited network bandwidth utilization and be enabled for autonomous AI updates as converged computing for different kinds of objects from different monitoring scenes to possible warning situations.

Our first contribution is that we propose and demonstrate a preliminary implementation of a newly developed distributed real-time object detection framework based on edge-cloud collaboration for smart video surveillance applications. Upon this framework, an edge-cloud collaboration mechanism is realized based on image matting and You Only Look Once (YOLO)-based AI methodology such as the state-of-the-art YOLOR object detection technique; where the cloud achieves global knowledge discovering from and enables global knowledge sharing to the edge to advance the whole framework for real-time smart video surveillance. The edge-cloud collaboration mechanism can autonomously update local AI models on edge devices having global knowledge for detecting and classifying unknown objects that appear frequently (objects are not seen beforehand). Lastly, the proposed framework can reduce/prevent network congestion (i.e., eliminate unnecessary data transmission) for real-time smart video surveillance when sending/transmitting a small set of images, with unknown latent objects, from the edge to the cloud when the network connectivity is available, compared to centralized cloud-centered paradigms.

The remainder of the paper is structured as follows. Section II describes the proposed distributed real-time object detection framework based on edge-cloud collaboration for real-time smart video surveillance applications, and its benefits. Section III describes the experiments that demonstrate a preliminary implementation of the proposed framework and validate its feasibility and effectiveness. Lastly, the conclusions are summarized in Section IV.

## II. METHODOLOGY

### A. FRAMEWORK

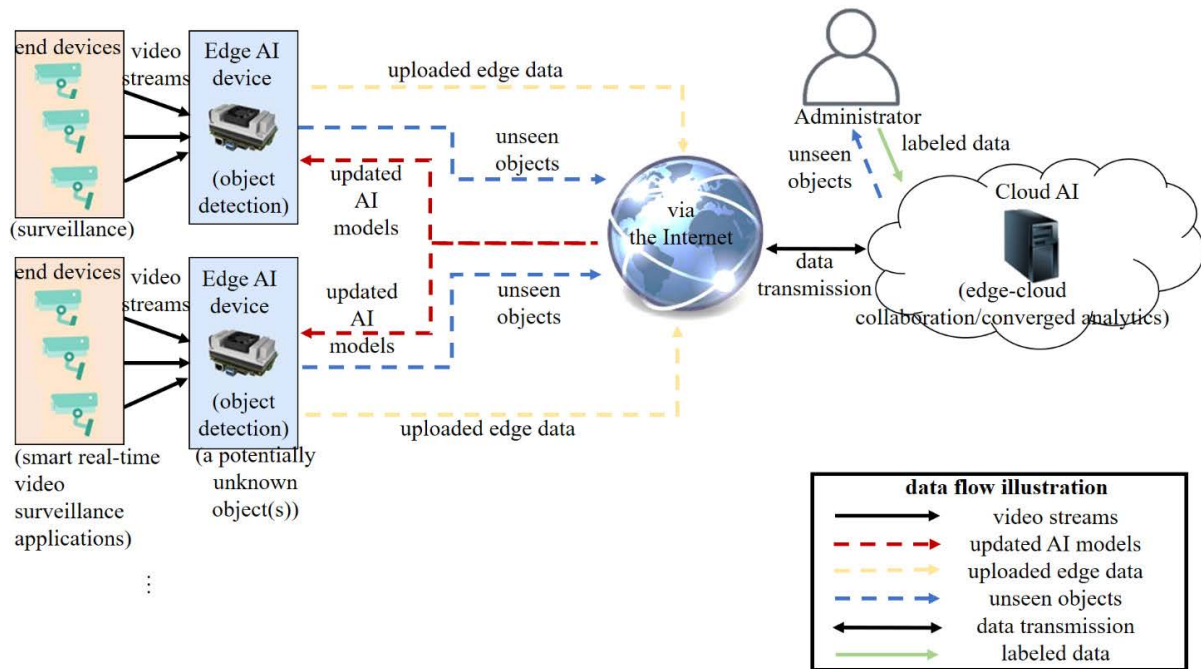
Figure 1 depicts the overview of the proposed distributed real-time object detection framework considering edge-cloud collaboration for real-time smart video surveillance at the edge. The framework is composed of three tiers: the end device, edge device and cloud tiers. The process flow for the proposed smart video surveillance based on edge-cloud collaboration is described as follows. IoT end devices - front-end video cameras - in the end device tier act as the first action that captures IoT data, i.e., media images in real time. The captured real-time media images are then passed to edge devices in the edge device tier. The edge devices connected to end devices and distributed geographically with storage and computing capabilities are responsible for processing and analyzing real-time media images for real-time smart video surveillance where they collaborate with the cloud in the cloud tier. The distributed edge devices for real-time captured,

processed and analyzed media images can be server-class machines or systems like Advanced RISC Machine (ARM)® processor-based embedded systems deployed on-site in different realistic fields of interest, which are with their trained local AI model for latency-sensitive video surveillance applications [26], [27]. When distributed on-site edge devices in the edge device tier detect unknown latent objects with a low confidence level for certain times, they transmit images to the cloud for object detection. In the cloud tier, the cloud having rich storage and computing resources to store captured image data from distributed on-site edge devices and consolidate stored image data through AI achieving global knowledge discovering assists distributed on-site edge devices to detect unknown latent objects. Also, trained local AI models - local inference models - to distributed on-site edge devices are updated remotely from the cloud enabling global knowledge sharing such that converged computing can be deployed to and performed on-line at the edge of the Internet. Such an edge-cloud collaboration mechanism where the cloud achieves global knowledge discovering from and enabling global knowledge sharing to the edge to advance the whole system for smart video surveillance is necessary because the performance and detection accuracy of trained local AI run on distributed edge devices for real-time video surveillance applications may be degraded over time with unknown objects. Additionally, global knowledge sharing can be triggered under a certain criterion that unknown objects frequently appeared for a certain number of times. In the cloud with such a mechanism, unknown latent objects related to a computed confidence level are found at the edge; further, queries in the form of requesting class labels for unlabeled images are sent over the Internet to an administrator overcoming the labeling bottleneck, where a practical paradigm of LINE Notify push notification service [14] is achieved. Owing to the distributed edge devices collaborating with the cloud, they can process and analyze captured real-time video images for real-time object detection without transmitting all image data to the cloud via the backbone Internet [28].

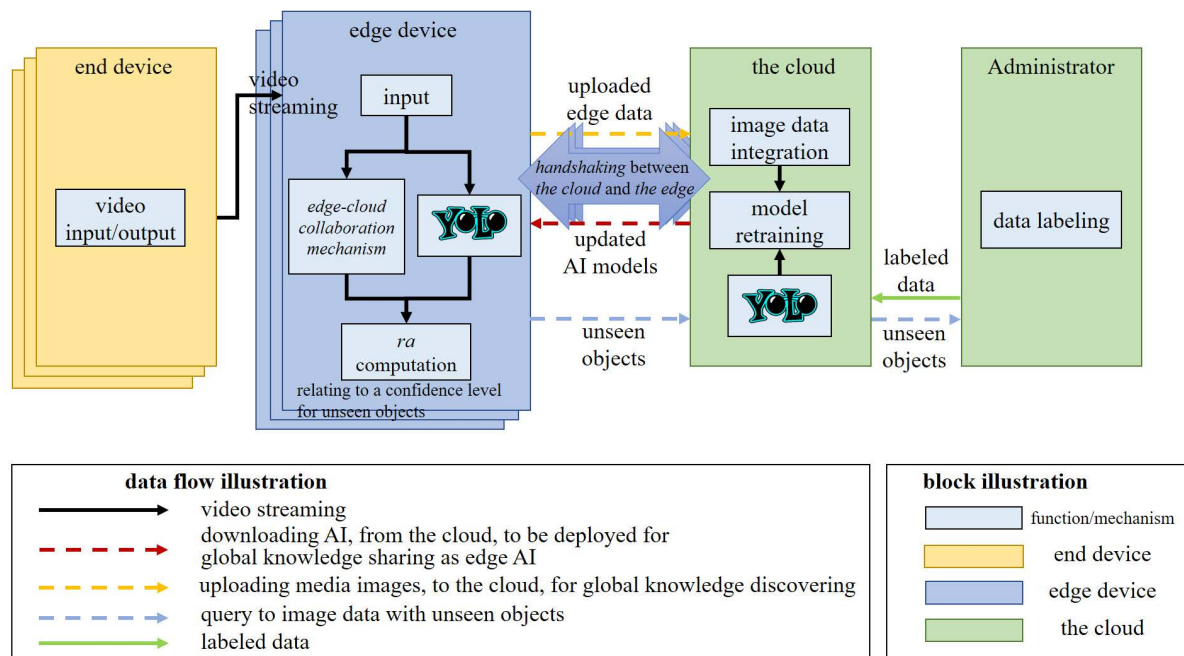
### B. PIPELINE OF DISTRIBUTED REAL-TIME OBJECT DETECTION UNDER EDGE-CLOUD COLLABORATION

Figure 2 illustrates the pipeline of distributed real-time object detection under edge-cloud collaboration in the proposed framework for real-time smart video surveillance at the edge.

In the pipeline, the end devices are responsible for real-time image data acquisition and transmission. Real-time video images are captured by end devices, and then, transmitted to on-site edge devices for further data processing and analysis. On-site edge devices receiving captured image data from end devices are responsible for real-time object detection and classification. They perform real-time object detection and classification based on their trained local one-stage object detection model, and collaborate with the cloud based on an edge-cloud collaboration mechanism when



**FIGURE 1.** An overview of the distributed real-time object detection framework considering edge-cloud collaboration proposed for real-time smart video surveillance at the edge.



**FIGURE 2.** A pipeline of distributed real-time object detection under edge-cloud collaboration in the framework proposed for real-time smart video surveillance at the edge. Numerous distributed on-site edge devices transmitting their video streams to and collaborating with the cloud to form a smart video surveillance orchestrator are the fundamental constituents of the whole system; they process and analyze video streams in real time for real-time smart video surveillance applications (the cloud is not suitable for tasks that are sensitive to network latency and congestion). The cloud uses stored video data, from numerous on-site edge devices distributed in various fields of interest at the edge of the Internet, to train, improve and optimize the model deployed remotely on distributed on-site edge devices. With an edge-cloud collaboration mechanism, the cloud can assist distributed on-site edge devices to update their inference model to detect unknown objects in images.

unknown objects exist in images and cannot be detected at the edge.

The edge-cloud collaboration mechanism in the pipeline of the proposed framework in this paper is described in



Section II-Part 1; the one-stage object detection approach is presented in Section II-Part 2. The cloud is responsible for image data storage and modeling via AI achieving global knowledge discovering and enabling global knowledge sharing. Through the edge-cloud collaboration mechanism, on-site edge devices, which serve as a promising complement of the cloud to support real-time smart video surveillance, collaborate with the cloud to refresh their knowledge (trained local AI model to be updated) in order to detect and classify unknown objects at the edge. The collaboration advances the whole system in real-time object detection and classification for smart real-time video surveillance applications.

### 1) EDGE-CLOUD COLLABORATION MECHANISM BASED UPON IMAGE MATTING AND YOLOV3 RELATING TO A CONFIDENCE LEVEL OF $ra$ FOR UNKNOWN LATENT OBJECTS

First, it is impractical to train a powerful AI model on an edge device to detect all objects for real-time video surveillance as an example. Practically, it can be alternated through an edge-cloud collaboration mechanism to adaptively update an AI model according to the environment where the edge device is located, as the cloud has large storage and computing resources for training a powerful AI model that can be deployed remotely for AI inference at the edge of the network (the Internet). Thus, in this paper, an edge-cloud collaboration mechanism developed and used to realize such an alternative is proposed. Figure 3 shows the block diagram of the proposed mechanism (which is mainly based upon image matting and YOLOv3, relating to a confidence level of residual area ( $ra$ ), for unknown latent objects to be learned for a new AI model trained in the cloud and deployed remotely on an edge device(s) for real-time smart video surveillance at the edge). We summarize the block diagram using an illustrative code snippet, as summarized in Table 1 (Algorithm 1).

The developed mechanism can assist an edge device(s) in adapting to the environment where it is located for real-time video surveillance at the edge. When an unknown latent object(s) appears frequently at the edge, the corresponding images are sent/transmitted to the cloud for off-line object detection and classification where an AI model is trained across all object classes. Once the cloud has trained a powerful AI model across all object classes (global knowledge discovering/converged analytics), it deploys/updates the trained model (via the network (the Internet)) with good generalization for AI inference at the edge (global knowledge sharing). With global knowledge discovering and sharing, the edge can know exactly if new objects are present or not. In the proposed distributed real-time object detection framework, the edge-cloud collaboration mechanism computes a confidence level of  $ra$  relating to a frequent occurrence to determine whether or not the cloud is requested by the edge to be assisted. Sending/transmitting a small set of images with unknown latent objects to the cloud, when  $ra$  is greater than  $\alpha$  for  $\beta$  times and the network is not congested, can reduce/prevent network congestion from the edge to the cloud with network

connectivity from the cloud to the edge for real-time smart video surveillance. Image matting and YOLOv3 relating to a confidence level of  $ra$  in the mechanism are briefly given below. Figure 4 shows the block diagram of image matting [29], i.e., a process of accurately estimating foreground objects in images, used in the edge-cloud collaboration mechanism above; this is a very important technique in image, video editing and composition applications. Foreground objects found in images through this process are considered and used against YOLOv3's object detection results to judge if the foreground objects are unknown in images at the edge or not (where  $ra$  is computed and compared as shown in Table 1 (Algorithm 1)). The output of each function block shown in Figure 4(a) is shown in Figure 4(b).

In Algorithm 1,  $ra$  is computed by Equation (1), where  $W$ ,  $H$  and  $A_i$  ( $n$  white pixels) are the width, height and white area of a resulting image through image cutting in Algorithm 1, respectively. If an unknown latent object(s) that cannot be detected and classified by YOLOv3 deployed at the edge and applied on a captured image (frame) exists, Equation (1) produces a very high value of  $ra$ ; otherwise, it produces a very low value of  $ra$ .

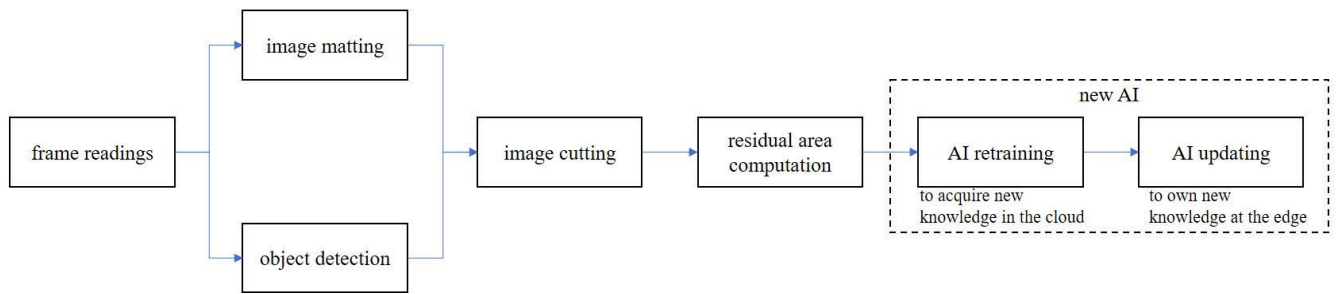
When an edge device(s) at the edge detects an unknown latent object(s) that frequently appeared for certain times with a confidence level of  $ra$  to be compared with a threshold, the cloud handshaking with the edge assists in detecting it.

$$ra = \frac{\sum_i^n A_i}{W \times H} \quad (1)$$

In the cloud with the same mechanism, unknown latent objects relating to a confidence level of  $ra$  are found at the edge; also, queries in the form of requesting class labels for unlabeled images are sent over the Internet to an administrator overcoming the labeling bottleneck, where a practical paradigm of LINE Notify push notification service is achieved.

### 2) ONE-STAGE OBJECT DETECTION APPROACH BASED ON YOLOV3

Convolutional Neural Networks (CNNs) are analogous to the organization and connectivity pattern of neurons in the visual cortex of the human brain [30]. CNN-based object detection approaches have been receiving a lot of attention from researchers for video surveillance applications, due to their superior performance; however, they can only be deployed on cloud machines having rich storage and computing resources [23]. For real-time object detection, one-stage object detection approaches like the YOLO series [31], [32], [33] are widely used. YOLO was first proposed in [24] to enhance the performance of two-stage object detection approaches like the faster Regional Convolutional Neural Network (R-CNN); R-CNNs are reliable, but they are slow even when they are running on a graphics processing unit (GPU). On the other hand, the one-stage object detection approach (YOLO) is very fast and can achieve very good real-time efficiency on a GPU [2]. It deals with



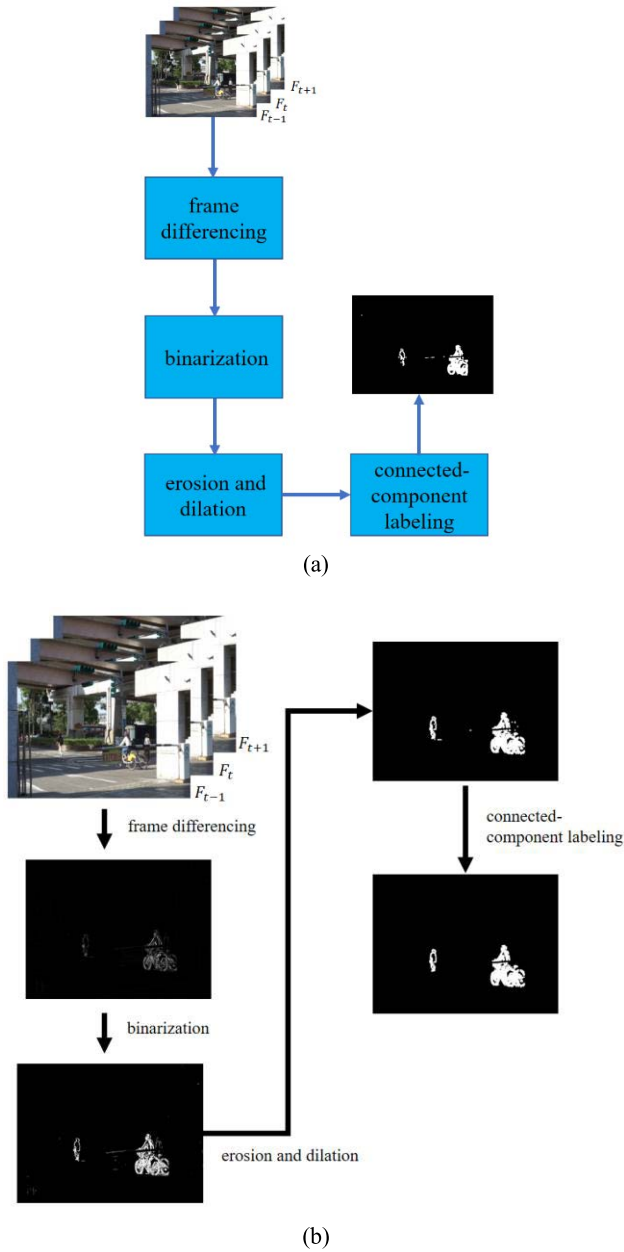
**FIGURE 3.** A block diagram of the proposed edge-cloud collaboration mechanism.

**TABLE 1.** Algorithm 1: Edge-cloud collaboration mechanism, based upon image matting and YOLOv3 relating to a confidence level of, for detecting unknown latent objects in real-time smart video surveillance at the edge. YOLO-accommodated AI methodology such as YOLOv4 and YOLOR against YOLOv3 can be considered; they are conducted and compared in this paper.

Algorithm 1: An edge-cloud collaboration mechanism for detecting unknown latent objects in real-time smart video surveillance at the edge
<b>Definitions:</b> $X_1$ : the present image frame; $X_2$ : the next image frame; $M_t$ : a matted image; $Y$ : YOLOv3's model; $\delta$ : bounding box coordinates by YOLOv3; $C_t$ : a resulting binary image of $M_t$ processed through image cutting with $\delta$ ; $\alpha$ : a threshold specified for a confidence level of $ra$ ; $\beta$ : a threshold specified for frequent occurrence of an unknown latent object(s); $c$ : a counter specified for $\beta$ , where its initial value is 0. <b>Inputs:</b> A frame sequence $\{F_1, F_2, \dots, F_{t-1}, F_t, F_{t+1}, \dots\}$ . <b>Outputs:</b> A set of images with an unknown latent object(s) $U_t$ . <b>While</b> Camera turned on <b>do</b> : <b>If</b> $F_t$ is the first frame <b>then</b> : $X_1 \leftarrow$ reading frame $F_t$ <b>else</b> : $X_2 \leftarrow$ reading frame $F_t$ $M_t \leftarrow$ image matting( $X_1, X_2$ ) $\delta \leftarrow$ YOLOv3 object detection( $X_2, Y$ ) $C_t \leftarrow$ image cutting( $M_t, \delta$ ) $ra \leftarrow$ residual area computation( $C_t$ ) <b>If</b> $ra > \alpha$ <b>then</b> : $U_t \leftarrow F_t$ $c \leftarrow c + 1$ <b>If</b> $c > \beta$ <b>and</b> the network is not congested <b>then</b> : sending images from the edge to the cloud( $U_t$ ) $c \leftarrow 0$ $U_t \leftarrow$ empty <b>End If</b> <b>End If</b> <b>If</b> the cloud, which has acquired new knowledge from $U_r$ , responds to the edge <b>then</b> : refreshing edge YOLOv3 $Y$ , which will own new knowledge at the edge <b>End If</b> $X_1 \leftarrow X_2$ <b>End If</b> $t \leftarrow t + 1$ <b>End While</b>

object detection as a regression problem, which is faster but less accurate than two-stage approaches like faster R-CNN [34], [35], [36]. Subsequently, the authors who proposed YOLO improved YOLOv1 to propose YOLOv2 [32]. YOLOv2 replaces the fully-connected layer of the YOLOv1 model with a fully convolutional layer, so as to have the

ability of handling images with different sizes. Also, the YOLOv1 model is upgraded in terms of object localization and multi-scale object detection. In 2018, YOLOv2 was improved and became YOLOv3 [33]. In comparison with YOLOv2, YOLOv3 greatly improves detection accuracy, while maintaining detection speed. YOLOv3 has



**FIGURE 4.** A flowchart of image matting used in the developed edge-cloud collaboration mechanism: (a) function blocks involved in image matting; (b) output of each function block.

been widely used in object detection owing to its excellent detection accuracy and speed [23], [33]. To achieve real-time video surveillance based on edge-cloud collaboration, AI deployed on edge devices should be robust without sacrificing detection accuracy over time while maintaining detection speed. In the pipeline of the proposed framework in this paper, real-time object detection and classification is achieved by YOLOv3 [37], which is conducted and used as a benchmark for the object detection method in the proposed distributed real-time object detection framework considering edge-cloud collaboration for real-time smart video surveillance applications.

The backbone of YOLOv3 has a total of 53 layers, and it uses the Residual Network (ResNet) to solve the problem of gradient vanishing. In addition, it uses the Feature Pyramid Network (FPN) [37] with multi-scale feature maps to detect objects with different sizes such that its performance in detecting small objects can be improved. Figure 5 shows the used YOLOv3 in this paper. Figure 5(a) shows that, it can use multi-scale feature maps, with different sizes, to strengthen its performance in detecting objects having different sizes; Figure 5(b) shows that, the feature maps with the different sizes are used to extract fine-grained, distinguishable features in distinguishing from objects having different sizes in an image.

In YOLOv3, the loss function of the model to be trained can be defined in Equation (2).

$$Loss = L_{bbox} + L_{obj} + L_{cls} \quad (2)$$

where  $L_{bbox}$  is the bounding box regression loss by the sum of squared error (SSE),  $L_{obj}$  represents the object confidence loss, and  $L_{cls}$  is the class score based on cross entropy (CE). The first loss ( $L_{bbox}$ ) in Equation (2) can be expressed as Equation (3).

$$L_{bbox} = \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^S \sum_{k=0}^B I_{ijk}^{obj} \times \sum_{r \in (x,y,w,h)} SSE(P_{bbox}^r, G_{bbox}^r) \quad (3)$$

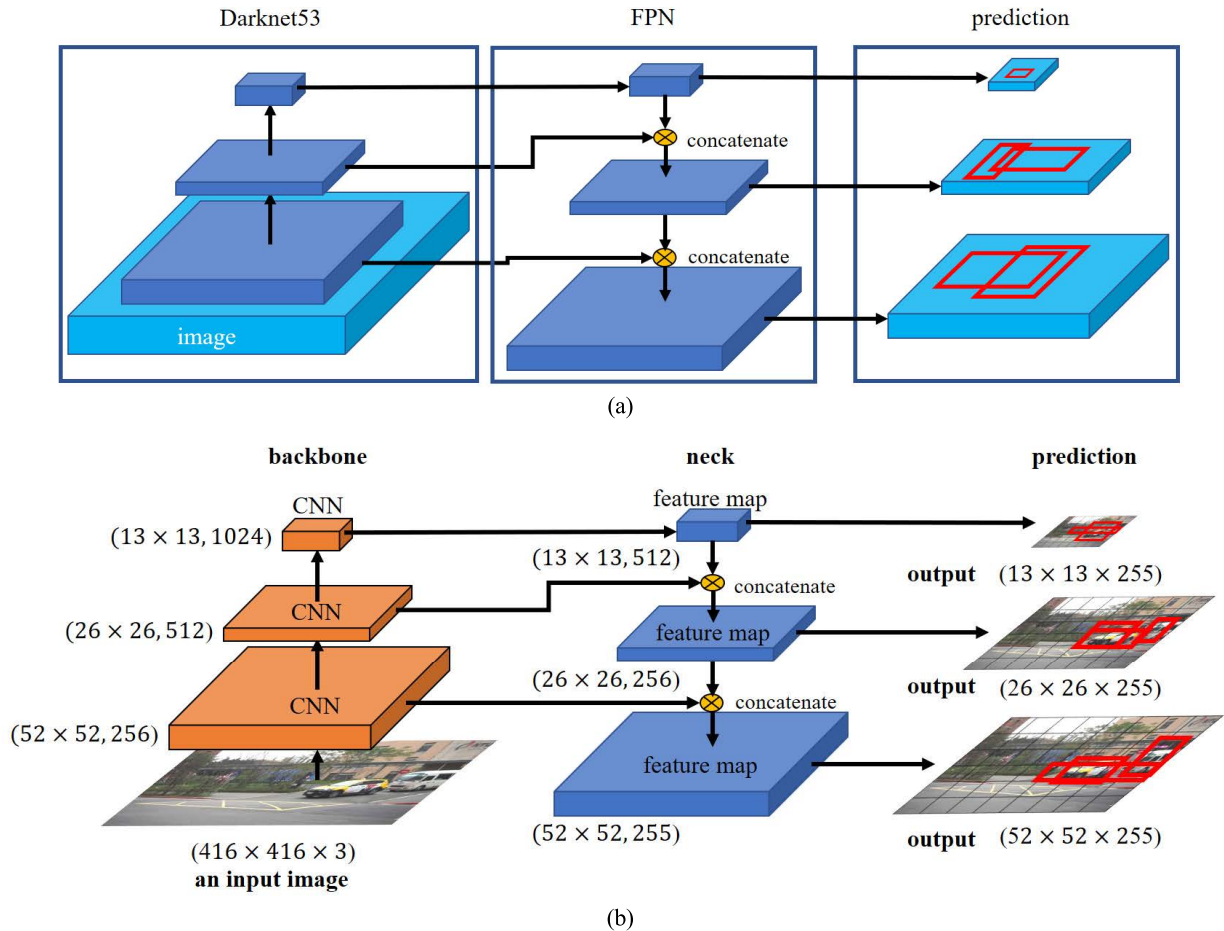
where  $P_{bbox}^r$  and  $G_{bbox}^r$  represent the predicted box and ground truth bounding box, respectively. Also,  $\lambda_{coord}$  indicates the loss weighting,  $S$  denotes the number of grids, and  $B$  accounts for the box generated by each grid cell.  $I_{ijk}^{obj}$  shows whether or not an object exists in the  $k^{th}$  grid cell ( $i, j$ ): if it exists,  $I_{ijk}^{obj} = 1$ ; otherwise, it is 0. In this paper, the Distance-Intersection over Union (DIoU) is considered and used, which is an improvement of the Intersection over Union (IoU)-based loss for object detection bounding box regression to its anchor boxes to be identified. In Equation (3),  $L_{bbox}$  is computed through the SSE between the predicted and ground truth bounding boxes.

The second loss ( $L_{obj}$ ) in Equation (2) can be expressed as in Equation (4).

$$L_{obj} = \sum_{i=0}^S \sum_{j=0}^S \sum_{k=0}^B I_{ijk}^{obj} SSE(P_{obj}, I^m) + \lambda_{noobj} \times \sum_{i=0}^S \sum_{j=0}^S \sum_{k=0}^B I_{ijk}^{noobj} SSE(P_{obj}, 0) \quad (4)$$

where  $P_{obj}$  is the probability that indicates whether or not an object exists in anchor boxes. The third loss ( $L_{cls}$ ) in Equation (2) can be expressed as in Equation (5).

$$L_{obj} = \sum_{i=0}^S \sum_{j=0}^S \sum_{k=0}^A I_{ijk}^{obj} \sum_{c=0}^C CE(P_{class}^c, G_{class}^c) \quad (5)$$



**FIGURE 5.** Used YOLOv3: (a) multi-scale feature maps with different sizes are used to strengthen the performance in detecting objects having different sizes; (b) image examples are shown, where the feature maps are with the different sizes.

where  $P_{class}^c$  represents the predicted class label  $c$  and  $G_{class}^c$  is the actual class label to be targeted.

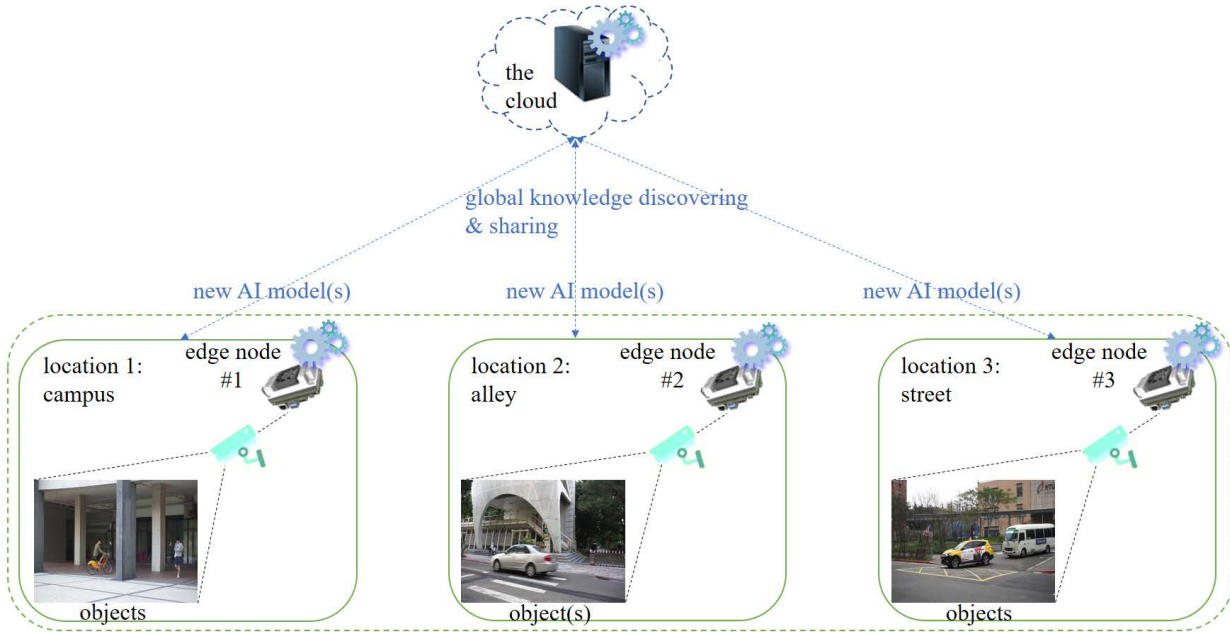
In this paper, state-of-the-art object detection techniques - You Only Learn One Representation (YOLOR) [38] and YOLOv4 [39] - are also used in the experiments in the following section and compared against YOLOv3 in terms of frames per second (FPS), model size (in MB) and mean Average Precision (mAP) as the evaluation metrics. FPS is used to evaluate the inference speed of the three state-of-the-art object detection techniques.

### III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the proposed distributed real-time object detection framework based on the edge-cloud collaboration mechanism for real-time smart video surveillance at the edge is experimentally evaluated. Figure 6 shows the preliminary implementation of the proposed framework for real-time smart video surveillance at the edge. The three distributed on-site edge devices connected with their video camera (their end device) are deployed at the three different locations, and the cloud is deployed in a laboratory environment. The distributed on-site edge devices are networked across wireless

Wi-Fi communication with the cloud. They transmit real-time video data to the cloud (based on the edge-cloud collaboration mechanism, they can send the cloud only a certain number of video data with potential new objects such that the bandwidth of the backbone network can be saved). They also process and analyze transmitted video data for real-time object detection and classification, using their trained local AI model. The cloud stores received video data from the distributed on-site edge devices. Further, the trained AI model (1) achieves global knowledge discovering from received video data captured at the different locations (for various scenarios) and (2) enables global knowledge sharing to the video data sources where distributed on-site edge devices refresh their knowledge (AI model) such that the whole video surveillance system is advanced with the edge-cloud collaboration mechanism for real-time smart video surveillance. Objects targeted by the distributed on-site edge devices with the focus on object detection and classification of real-time smart video surveillance potential are presented in Table 2. The objects include pedestrians, bicycles, cars, motorcycles, trucks and buses; they are large objects and some of them are potentially dangerous targets (forbidden or restricted) for certain areas





**FIGURE 6.** Preliminary implementation of the proposed distributed real-time object detection framework based on the edge-cloud collaboration mechanism for real-time smart video surveillance at the edge. In the proposed framework, the distributed on-site edge devices connected with their end device and deployed at different locations can transmit different real-time video data to the cloud, and the cloud can optimize AI based on received video data. The edge-cloud collaboration mechanism can advance the whole system.

**TABLE 2.** Objects targeted by the distributed on-site edge devices.

		class					
		pedestrian	bicycle	car	motorcycle	truck	bus
the edge	edge node #1	✓	×	×	×	×	×
	edge node #2	✓	✓	✓	✓	×	×
	edge node #3	✓	✓	✓	✓	✓	✓

of interest. Table 3 lists the specifications of hardware and software used in the preliminary implementation.

In the preliminary implementation, three power-efficient and compact nVIDIA® Jetson Xavier™ NX modules are used as the distributed on-site edge devices performing deployed AI updated from the cloud for real-time smart video surveillance applications. The nVIDIA® Jetson Xavier™ NX benefits from new cloud-native support, and accelerates the nVIDIA software stack in as little as 10 W with more than 10 times the performance of its widely adopted predecessor Jetson™ TX2. With regards to the cloud, the nVIDIA® GeForce™ RTX 2080Ti GPU is configured on a computer with an Intel® Core™ i7-9700k CPU (@ 4.9GHz), and it is used to model AI from gathered image data, to be used as converged computing, deployed on-site and performed at the edge. Here, AI trained in the cloud and deployed at the edge is based on YOLOv3, which is implemented in PyTorch (based on Python). In addition, labeled image data from the COCO 2017 dataset [40] are used; a total

of 4,004 images are randomly selected for training, while the remaining 1,001 images are used for testing. When collaborating with the cloud (which enables global knowledge sharing based on the image matting-based edge-cloud collaboration mechanism), the distributed on-site edge devices are capable of identifying unknown latent objects using their local updated AI model (in the preliminary implementation of the proposed framework, each on-site edge device owns its trained local AI model identifying certain objects for its specific task, while the cloud receiving video streams from all the on-site edge devices leverages them to serve as converged computing to be deployed and performed at the edge). The trained local AI models that did not include unknown objects are updated, when unknown objects frequently appeared for certain times that are related to a confidence level of  $ra$ . Here, the unknown objects frequently appeared for  $\beta = 10$  times at which  $ra$  is greater than or equal to  $\alpha = 1.0\%$ . The results of the edge-cloud collaboration mechanism based on image matting with  $ra$  for unknown latent objects can be seen in

**TABLE 3.** Specifications of hardware and software used in the preliminary implementation.

	device	hardware	software
end device	camera	Logitech webcam C920	-
edge device	embedded system	nVIDIA® Jetson Xavier™ NX	YOLOv3 (AI inference)
cloud server	PC	<ul style="list-style-type: none"> <li>CPU: Intel® Core™ i7-9700K (RAM: 32GB)</li> <li>GPU: GeForce RTX™ 2080 Ti</li> </ul>	YOLOv3 (AI training)

**Algorithm 2** Implementation of AI by YOLOv3 That is Trained in the Cloud and Deployed at the Edge




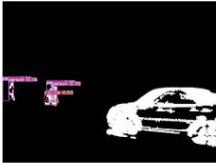

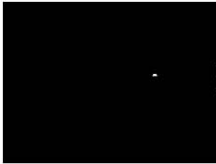
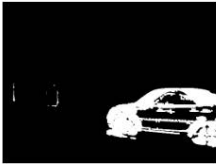
1. Inputs ( $416 \times 416, 3$ )
2. CBL(Conv2d(size =  $3 \times 3/2$ ) Batch Normalization, and Leaky ReLU,  $416 \times 416, 32$ )
3.  $1 \times$  Res Block(CBL(size =  $3 \times 3/2$ ) and ResNet,  $208 \times 208, 64$ )
4.  $2 \times$  ResBlock(CBL(size =  $3 \times 3/2$ ) and ResNet,  $104 \times 104, 128$ )
5.  $8 \times$  Res Block(CBL(size =  $3 \times 3/2$ ) and ResNet,  $52 \times 52, 256$ )  $\rightarrow$  line 13
6.  $8 \times$  Res Block(CBL(size =  $3 \times 3/2$ ) and ResNet,  $26 \times 26, 512$ )  $\rightarrow$  line 10
7.  $4 \times$  Res Block(CBL(size =  $3 \times 3/2$ ) and ResNet,  $13 \times 13, 1024$ )
8. CBL Set(CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ),  $13 \times 13, 512$ )  $\rightarrow$  CBL ((size =  $3 \times 3$ ),  $13 \times 13, 255$ ) + Conv2d ((size =  $1 \times 1$ )  $13 \times 13, 255$ )  $\rightarrow$  **Output prediction**
9. CBL(size =  $1 \times 1$ ) + up sampling( $26 \times 26, 256$ )
10. concatenate( $26 \times 26, 768$ ) with line 6.
11. CBL Set(CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ),  $26 \times 26, 256$ )  $\rightarrow$  CBL ((size =  $3 \times 3$ ),  $26 \times 26, 255$ ) + Conv2d ((size =  $1 \times 1$ )  $26 \times 26, 255$ )  $\rightarrow$  **Output prediction**
12. CBL(size =  $1 \times 1$ ) + up sampling( $52 \times 52, 128$ )
13. concatenate( $52 \times 52, 384$ ) with line 5.
14. CBL Set(CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ), CBL(size =  $3 \times 3$ ), CBL(size =  $1 \times 1$ ),  $52 \times 52, 255$ )  $\rightarrow$  CBL ((size =  $3 \times 3$ ),  $52 \times 52, 255$ ) + Conv2d ((size =  $1 \times 1$ ),  $52 \times 52, 255$ )  $\rightarrow$  **Output prediction**

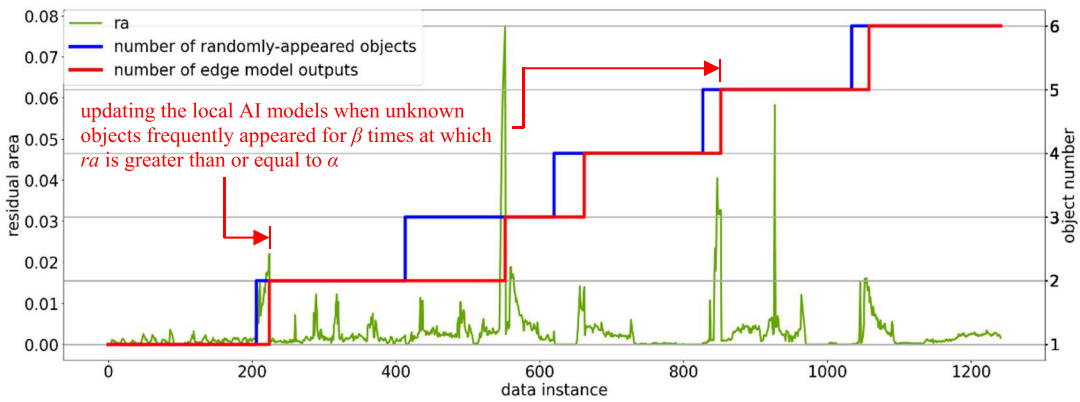
Table 4. In Table 4 (a) original images with objects are shown; in Table 4 (b) objects are detected through YOLOv3 object detection; in Table 4 (c) matted images with bounding boxes by YOLOv3 for objects detected are shown (original images are processed through image matting); and in Table 4 (d) unknown objects may exist with  $ra (\geq 1.0(\%))$  (matted images are processed via image cutting), which can be identified by updated inference models at the edge. In the cloud with the same mechanism, unknown latent objects relating to  $ra$  are found at the edge, and queries in the form of requesting class labels for unlabeled images are sent over the Internet to an administrator overcoming the labeling bottleneck; a practical paradigm of LINE Notify push notification service is implemented.

Figure 7 shows the detection of unknown objects at the edge. The blue line indicates the situations in which unseen types of objects frequently appeared, while the red line denotes the situations in which unseen objects that frequently appeared (for 10 times) are identified successfully after the trained local models on the distributed on-site edge devices are autonomously updated (the edge devices transmit a small set of unknown latent object-contained images to the cloud, while the cloud achieves global knowledge discovering from

received images and enables global knowledge sharing to deploy the model at the edge). As seen in Figure 7 that shows the interaction between the cloud and the edge for the model updates in the proposed edge-cloud collaborative framework, the local AI models are updated when unknown objects frequently appeared for  $\beta = 10$  times when  $ra$  (computed by Equation (1)) is greater than or equal to  $\alpha = 0.01$ . Once the local AI models have been updated upon the proposed edge-cloud collaboration mechanism, the edge devices have the ability of identifying the never-before-seen objects. This is evidenced by  $ra$  that returns to the normal level, as shown in Figure 7. The autonomous update of the trained local AI models from the cloud to the edge through the proposed edge-cloud collaboration mechanism is realized; thus, the edge-cloud collaboration advances the whole system. With the proposed mechanism, the edge devices can transmit a small set of unknown latent object-contained images to the cloud. Figure 8 shows the bandwidth occupancy of the cloud against the edge. As shown in the figure, the proposed edge-cloud collaborative framework could save approximately 94% bandwidth of the backbone network. Based on this mechanism, the distributed on-site edge devices need to transmit only a small number of unknown latent object-contained

**TABLE 4.** Results of the edge-cloud collaboration mechanism, Algorithm 1, based on image matting and object detection with a confidence level of  $ra$  for unknown latent objects to be learned for a new AI model trained in the cloud and then deployed at the edge for real-time smart video surveillance.

edge location	campus	street	alley
(a) original image			
(b) YOLOv3 object detection			
(c) image matting			
(d) image cutting			
$ra$ (%)	0.05	0.03	10.82



**FIGURE 7.** Detection of unknown objects at the edge ( $\alpha$  specified for  $ra$ : 1.0%;  $\beta$  specified for frequent occurrence of an unknown latent object relating to  $ra$ : 10).

images to the cloud building new global AI for the model updates shown in Figure 7.

To diminish the risk of overfitting that the AI model (YOLOv3) trained in the cloud consolidates

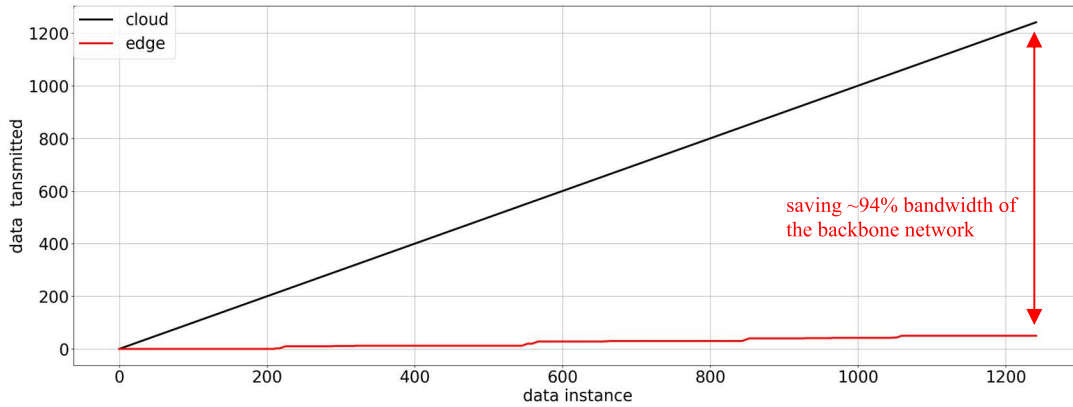


FIGURE 8. Bandwidth occupancy of the cloud against the edge.

TABLE 5. Results of the AI model that serves as converged computing at the edge for real-time smart video surveillance.

		mAP@.5					
		class					
		pedestrian	bicycle	car	motorcycle	truck	bus
model	model 1	0.884	×	×	×	×	×
	model 2	0.907	0.577	×	×	×	×
	model 3	0.885	0.883	0.949	×	×	×
	model 4	0.9	0.913	0.969	0.965	×	×
	model 5	0.919	0.909	0.963	0.976	0.904	×
	model 6	0.897	0.95	0.989	0.92	0.803	0.969

Note: Stratified  $k$ -fold cross-validation test: for each stratified fold, the percentage of data instances in each target class is approximately the same.

class-imbalanced image data from the distributed on-site edge devices, stratified  $k$ -fold cross-validation test [41], [42], [43] (i.e., a variant of the commonly used  $k$ -fold cross-validation strategy) evaluates the AI model. First, it returns  $k$  stratified folds ( $k = 5$  here) from the entire dataset split. Then, the following procedure is followed in a loop for each of the  $k$  stratified folds: (1) the AI model is trained on  $k - 1$  of the stratified folds as the training data, and (2) the trained model is validated on a remaining stratified one as the test set to compute the performance measure (i.e., mAP@.5) [44]. Finally, the overall performance measure reported by the stratified  $k$ -fold cross-validation test is the average of the measures computed in the loop. In the stratified  $k$ -fold cross-validation test, the stratification ensures that, as the complete set, each stratified fold contains approximately the same percentage of data instances of each target class (the stratified  $k$ -fold cross-validation test preserves approximate class ratios in both training and test datasets).

Table 5 shows the results of the AI model that serves as converged computing at the edge for real-time smart video

surveillance. Table 6 shows the results of the AI model trained through the holdout cross-validation test (Table 6 (a)) versus the stratified  $k$ -fold cross-validation test (Table 6 (b)), where  $F_1$ -score was used as the metric; only four classes were considered and shown in Table 6 (a). As shown in Table 5, the final AI model (model 6) achieves an excellent level of detection accuracy of 0.921 across all the six object classes, which gives a value of  $F_1$  score of 0.812 across them as shown in Table 6 (b). Table 7 presents the evaluation metrics from the comparison of the three state-of-the-art object detection techniques in terms of mAP@.5, model size and FPS. As shown in Table 7, serving as edge computing, YOLOv3 obtains an FPS value of 15.7 under an achieved rate of up to 1.44 and 1.33 times to that of YOLOv4 and YOLOR, respectively. Also, YOLOv3 is a lightweight model with a model size of 453.32 MB, which shrinks by approximately 69% and 61% against the model size of 1465.36 MB by YOLOv4 and model size of 1163.73 MB by YOLOR, respectively. In summary, YOLOv3 is chosen and used as the primary AI technique for the edge devices in the proposed framework



**TABLE 6.** Results of the AI model trained with (a) a holdout cross-validation test, (b) a stratified k-fold cross-validation test.

		(A)					
		F <sub>1</sub> score					
		class					
		pedestrian	bicycle	car	motorcycle		
model	model 1	0.484	✖	✖	✖		
	model 2	0.652	0.25	✖	✖		
	model 3	0.525	0.224	0.564	✖		
	model 4	0.57	0.211	0.794	0.873		
	model 5	0.62	0.239	0.702	0.874		
	model 6	0.499	0.254	0.879	0.906		

		(B)					
		F <sub>1</sub> score					
		class					
		pedestrian	bicycle	car	motorcycle	truck	bus
model	model 1	0.709	✖	✖	✖	✖	✖
	model 2	0.724	0.615	✖	✖	✖	✖
	model 3	0.758	0.914	0.648	✖	✖	✖
	model 4	0.795	0.923	0.747	0.910	✖	✖
	model 5	0.740	0.858	0.750	0.908	0.806	✖
	model 6	0.721	0.916	0.758	0.838	0.764	0.874

**TABLE 7.** Evaluation metrics of the three state-of-the-art object detection techniques.

object detection technique	backbone	mAP@.5 (across all the classes)	model size (MByte)	FPS (on GeForce RTX™ 2080 Ti)	FPS (on Jetson Xavier™ NX)
YOLOv3	Darknet53	0.944	453.32	103	15.7
YOLOv4	CSPDarknet53	0.983	1465.36	61.3	10.9
YOLOR	Yolor_CSP	0.977	1163.73	61.7	11.8

as it reaches the minimum model size to be deployed and achieves the highest FPS, where the similar performance in mAP@.5 is given by it; it throwing a value of mAP@.5 of 0.944 across all the six object classes achieves an excellent level of detection accuracy. Through collaborating with the cloud, the distributed on-site edge devices identifying objects in real time can identify specific target objects or potentially dangerous targets forbidden in certain restricted areas for real-time smart video surveillance. As demonstrated in this section, with the edge-cloud collaboration mechanism, the cloud assists on-site edge devices to detect unknown objects, in images, using their updated inference model from the cloud.

#### IV. CONCLUSION AND FUTURE WORK

For the development of smart video surveillance applications, edge computing, which is a promising complement of the

cloud, has pushed media data processing from the cloud to the edge of the Internet, and can achieve fast response for latency-sensitive video surveillance tasks. In this paper, a distributed real-time object detection framework based on edge-cloud collaboration for smart video surveillance applications has been proposed. It achieves fast response for real-time video surveillance while serving as converged computing by which media data from distributed edge devices are consolidated through AI in the cloud; AI achieving global knowledge discovering in the cloud is then deployed, as global knowledge sharing, remotely on distributed edge devices. Additionally, its preliminary implementation has been demonstrated and validated experimentally. In the future, a peer-to-peer offloading mechanism [22], [25] among the edge devices, to balance real-time workloads, for distributed/decentralized computing in the proposed framework for smart video surveillance (a video-/image-processing related practical application) will

be developed. The framework proposed in this paper is suitable for internal networks where data security and privacy protection are not considered. In the future, some of the main cryptosystems, such as proxy re-encryption, homomorphic encryption and searchable encryption [12], [45], [46], [47], [48], that can be implemented for data security and privacy protection will be investigated and included in the proposed framework for networks that require them.

## REFERENCES

- [1] Y.-Y. Chen, M.-H. Chen, C.-M. Chang, F.-S. Chang, and Y.-H. Lin, "A smart home energy management system using two-stage non-intrusive appliance load monitoring over fog-cloud analytics based on Tridium's Niagara framework for residential demand-side management," *Sensors*, vol. 21, no. 8, p. 2883, Apr. 2021.
- [2] M. A. Ezzat, M. A. A. E. Ghany, S. Almotairi, and M. A.-M. Salem, "Horizontal review on video surveillance for smart cities: Edge devices, applications, datasets, and future trends," *Sensors*, vol. 21, no. 9, p. 3222, May 2021.
- [3] S. S. Ahamad and A.-S. K. Pathan, "A formally verified authentication protocol in secure framework for mobile healthcare during COVID-19-like pandemic," *Connection Sci.*, vol. 33, no. 3, pp. 532–554, Jul. 2021.
- [4] A. Song and M. Zhang, "Genetic programming for detecting target motions," *Connection Sci.*, vol. 24, nos. 2–3, pp. 117–141, Sep. 2012.
- [5] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov./Dec. 2018.
- [6] Y. Zhang, J. Ren, J. Liu, C. Xu, H. Guo, and Y. Liu, "A survey on emerging computing paradigms for big data," *Chin. J. Electron.*, vol. 26, no. 1, pp. 1–12, Jan. 2017.
- [7] H. Zhao, L. Yao, Z. Zeng, D. Li, J. Xie, W. Zhu, and J. Tang, "An edge streaming data processing framework for autonomous driving," *Connection Sci.*, vol. 33, no. 2, pp. 173–200, Apr. 2021.
- [8] Q. Xu, Z. Su, Q. Zheng, M. Luo, and B. Dong, "Secure content delivery with edge nodes to save caching resources for mobile users in green cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2550–2559, Jun. 2018.
- [9] T. Sultana and K. A. Wahid, "IoT-guard: Event-driven fog-based video surveillance system for real-time security management," *IEEE Access*, vol. 7, pp. 134881–134894, 2019.
- [10] T. Sultana and K. A. Wahid, "Choice of application layer protocols for next generation video surveillance using internet of video things," *IEEE Access*, vol. 7, pp. 41607–41624, 2019.
- [11] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
- [12] X. Yan, P. Yin, Y. Tang, and S. Feng, "Multi-keywords fuzzy search encryption supporting dynamic update in an intelligent edge network," *Connection Sci.*, vol. 34, no. 1, pp. 511–528, Dec. 2022.
- [13] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Aug. 2017.
- [14] Y.-Y. Chen, Y. H. Lin, C. C. Kung, M. H. Chung, and I. H. Yen, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors*, vol. 19, no. 9, p. 2047, 2019.
- [15] D. Aishwarya and R. I. Minu, "Edge computing based surveillance framework for real time activity recognition," *ICT Exp.*, vol. 7, no. 2, pp. 182–186, Jun. 2021.
- [16] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.
- [17] T. Abdullah, A. Anjum, M. F. Tariq, Y. Baltaci, and N. Antonopoulos, "Traffic monitoring using video analytics in clouds," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput.*, Dec. 2014, pp. 39–48.
- [18] A. Anjum, T. Abdullah, M. F. Tariq, Y. Baltaci, and N. Antonopoulos, "Video stream analysis in clouds: An object detection and classification framework for high performance video analytics," *IEEE Trans. Cloud Comput.*, vol. 7, no. 4, pp. 1152–1167, Oct. 2019.
- [19] M. A. Hossain, "Framework for a cloud-based multimedia surveillance system," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 5, p. 135257, May 2014.
- [20] L. Valentin, S. A. Serrano, R. O. García, A. Andrade, M. A. Palacios-Alonso, and L. E. Súcar, "A cloud-based architecture for smart video surveillance," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-4/W3, pp. 99–104, 2017.
- [21] C.-S. Yoon, H.-S. Jung, J.-W. Park, H.-G. Lee, C.-H. Yun, and Y. W. Lee, "A cloud-based UTOPIA smart video surveillance system for smart cities," *Appl. Sci.*, vol. 10, no. 18, p. 6572, Sep. 2020.
- [22] Q. Zhang, Z. Yu, W. Shi, and H. Zhong, "Demo abstract: EVAPS: Edge video analysis for public safety," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 121–122.
- [23] Z. Xu, J. Li, and M. Zhang, "A surveillance video real-time analysis system based on edge-cloud and FL-YOLO cooperation in coal mine," *IEEE Access*, vol. 9, pp. 68482–68497, 2021.
- [24] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4962–4974, Aug. 2020.
- [25] R. Rajavel, S. K. Ravichandran, K. Harimoorthy, P. Nagappan, and K. R. Gobichettipalayam, "IoT-based smart healthcare video surveillance system using edge computing," *J. Ambient Intell. Hum. Comput.*, vol. 13, no. 6, pp. 3195–3207, Jun. 2022.
- [26] L. Duan, Y. Lou, S. Wang, W. Gao, and Y. Rui, "AI-oriented large-scale video management for smart city: Technologies, standards, and beyond," *IEEE Multimedia Mag.*, vol. 26, no. 2, pp. 8–20, Apr. 2019.
- [27] Y.-D. Zhang, Z.-J. Yang, H.-M. Lu, X.-X. Zhou, P. Phillips, Q.-M. Liu, and S.-H. Wang, "Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation," *IEEE Access*, vol. 4, pp. 8375–8385, 2016.
- [28] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.
- [29] M. B. Dillencourt, H. Samet, and M. Tamminen, "A general approach to connected-component labeling for arbitrary image representations," *J. ACM*, vol. 39, no. 2, pp. 253–280, Apr. 1992.
- [30] E. B. Varghese and S. M. Thampi, "Towards the cognitive and psychological perspectives of crowd behaviour: A vision-based analysis," *Connection Sci.*, vol. 33, no. 2, pp. 380–405, Apr. 2021.
- [31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [32] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
- [33] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.
- [34] D. Acharya, K. Khoshelham, and S. Winter, "Real-time detection and tracking of pedestrians in CCTV images using a deep convolutional neural network," in *Proc. Therapeutic 4th Annu. Conf. Research@Locate*, 2017, pp. 3–6.
- [35] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [37] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
- [38] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," 2021, *arXiv:2105.04206*.
- [39] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, 2014, pp. 740–755.
- [41] P. Dhivya and S. Vasuki, "Wavelet based MRI brain image classification using radial basis function in SVM," in *Proc. 2nd Int. Conf. Trends Electron. Informat. (ICOET)*, May 2018, pp. 1–9.
- [42] E. M. Dogo, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, "Empirical comparison of approaches for mitigating effects of class imbalances in water quality anomaly detection," *IEEE Access*, vol. 8, pp. 218015–218036, 2020.
- [43] D. Zhang, Z. Chen, M. K. Awad, N. Zhang, H. Zhou, and X. S. Shen, "Utility-optimal resource management and allocation algorithm for energy harvesting cognitive radio sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3552–3565, Dec. 2016.
- [44] Z. Song, J. Yang, D. Zhang, S. Wang, and Z. Li, "Semi-supervised dim and small infrared ship detection network based on Haar wavelet," *IEEE Access*, vol. 9, pp. 29686–29695, 2021.

- [45] R. Gupta and U. P. Rao, "A hybrid location privacy solution for mobile LBS," *Mobile Inf. Syst.*, vol. 2017, pp. 1–11, Jun. 2017.
- [46] L. Wu, X. Wei, L. Meng, S. Zhao, and H. Wang, "Privacy-preserving location-based traffic density monitoring," *Connection Sci.*, vol. 34, no. 1, pp. 874–894, Dec. 2022.
- [47] P. Yang, X. Gui, J. An, and F. Tian, "An efficient secret key homomorphic encryption used in image processing service," *Secur. Commun. Netw.*, vol. 2017, pp. 11–13, Mar. 2017.
- [48] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.



**YUNG-YAO CHEN** (Member, IEEE) received the B.S. and M.S. degrees in electrical and control engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from Purdue University, USA, in 2013. Before being a Faculty, he worked with HP Labs–Printing and Content Delivery Laboratory (HPL–PCDL) about one year. He is currently an Associate Professor with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include vision-based automation, automated/wisdom factory, self-driving car, and human–computer interaction. He is a member of the Golden Key International Honor Society and Phi Tau Phi. He was a recipient of the Ta-Yu Wu Memorial Award from Taiwan's Ministry of Science and Technology (MOST).



**YU-HSIU LIN** received the Ph.D. degree in mechanical and electrical engineering from the Graduate Institute of Mechanical and Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, in 2014. To his work and research experience, from October 2014 to August 2017, he worked with the Smart Network System Institute, Institute for Information Industry (III), Taiwan, and worked as a full-time Senior Engineer. From September 2017 to July 2018, he was an Assistant Professor with the Department of Computer Science and Information Management, Providence University, Taichung City, Taiwan. From August 2018 to July 2019, he was an Assistant Professor with the Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Tainan City, Taiwan. Besides, from August 2019 to January 2021, he was an Assistant Professor with the Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City, Taiwan. Since February 2021, he has been with the Graduate Institute of Automation Technology, National Taipei University of Technology, where he is currently an Assistant Professor. His current research interests include the IoT technologies, fog/edge-cloud computing, and artificial intelligence/deep learning/computational intelligence in smart grid.



**YU-CHEN HU** (Senior Member, IEEE) received the Ph.D. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, in 1999. He is currently a Professor with the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. His research interests include image and signal processing, data compression, information hiding, information security, computer networks, and machine learning.



**CHIH-HSIEN HSIA** (Member, IEEE) was born in Taipei City, Taiwan, in 1979. He received the Ph.D. degree from Tamkang University, New Taipei, Taiwan, in 2010.

In 2007, he was a Visiting Scholar with Iowa State University, Ames, IA, USA. From 2010 to 2013, he was a Postdoctoral Research Fellow with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei. From 2013 to 2015, he was an Assistant Professor with the Department of Electrical Engineering, Chinese Culture University, Taiwan. He was an Associate Professor with the Chinese Culture University and the National Ilan University, Taiwan, from 2015 to 2017. From 2019 to 2020, he was the Director of the Research Planning Division, Research & Development, National Ilan University. He is currently a Professor and the Chairperson with the Department of Computer Science and Information Engineering, National Ilan University. He is also the Director of the Multimedia & Intelligent Technical Laboratory, National Ilan University. His research interests include DSP IC Design, AI in computer vision, and cognitive learning. He was received the Outstanding Young Scholar Award of the Taiwan Association of Systems Science and Engineering in 2020 and the Outstanding Young Scholar Award of the Computer Society of the Republic of China in 2018. He is the Chapter Chair of the IEEE Young Professionals Group, Taipei Section, and the Director of the IET Taipei Local Network. He has served as an Associate Editor for the *Journal of Imaging Science and Technology*, the *Journal of Imaging*, and the *Journal of Computers*.



**YI-AN LIAN** received the B.S. degree in electronic engineering from the National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, in 2019. He is currently pursuing the M.S. degree in electronic and computer engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include digital image processing and deep learning.



**SIN-YE JHONG** received the B.S. degree in electrical engineering from Chinese Culture University, Taipei, Taiwan, in 2017, and the M.E. degree from the Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, in 2019. He is currently pursuing the Ph.D. degree with the Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan. His research interests include digital image and video processing, computer vision, and deep learning. He is also the Vice Chair of Young Professionals with the IET Taipei Local Network.

...