AI-Enabled IIoT for Live Smart City Event Monitoring

Md Abdur Rahman, *Senior Member, IEEE*, M. Shamim Hossain, *Senior Member, IEEE*, Ahmad J. Showail, Nabil A. Alrajeh, and Ahmed Ghoneim, *Senior Member, IEEE*

***Abstract*—Recent advancements of the Industrial Internet of Things (IIoT) have revolutionized modern urbanization and smart cities. While IIoT data contain rich events and objects of interest, processing a massive amount of IIoT data and making predictions in real-time are challenging. Recent advancements in artificial intelligence (AI) allow processing such a massive amount of IIoT data and generating insights for further decision-making processes. In this article, we propose several key aspects of AI-enabled IIoT data for smart city monitoring. First, we have combined a human-intelligence-enabled crowdsourcing application with that of an AI-enabled IIoT framework to capture events and objects from IIoT data in real time. Second, we have combined multiple AI algorithms that can run on distributed edge and cloud nodes to automatically categorize the captured events and objects and generate analytics, reports, and alerts from the IIoT data in real time. The results can be utilized in two scenarios. In the first scenario, the smart city authority can authenticate the AI-processed events and assign these events to the appropriate authority for managing the events. In the second scenario, the AI algorithms are allowed to interact with humans or IIoT for further processes. Finally, we will present the implementation details of the scenarios mentioned above and the test results. The test results show that the framework has the potential to be deployed within a smart city.**

***Index Terms*—Artificial intelligence (AI), cyber–physical systems, Industrial Internet of Things (IIoT).**

Manuscript received 4 April 2021; revised 30 May 2021, 20 June 2021, and 11 August 2021; accepted 26 August 2021. Date of publication 1 September 2021; date of current version 6 February 2023. This work was supported by the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia, through the Research Group Project under Grant RGP-229. *(Corresponding author: M. Shamim Hossain.)*

Md Abdur Rahman is with the Department of Cyber Security and Forensic Computing, College of Computer and Cyber Sciences, University of Prince Mugrin, Madinah 41499, Saudi Arabia (e-mail: m.arahman@upm.edu.sa).

M. Shamim Hossain is with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mshossain@ksu.edu.sa).

Ahmad J. Showail is with the Department of Computer Engineering, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia (e-mail: ashowail@taibahu.edu.sa).

Nabil A. Alrajeh is with the Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: nabil@ksu.edu.sa).

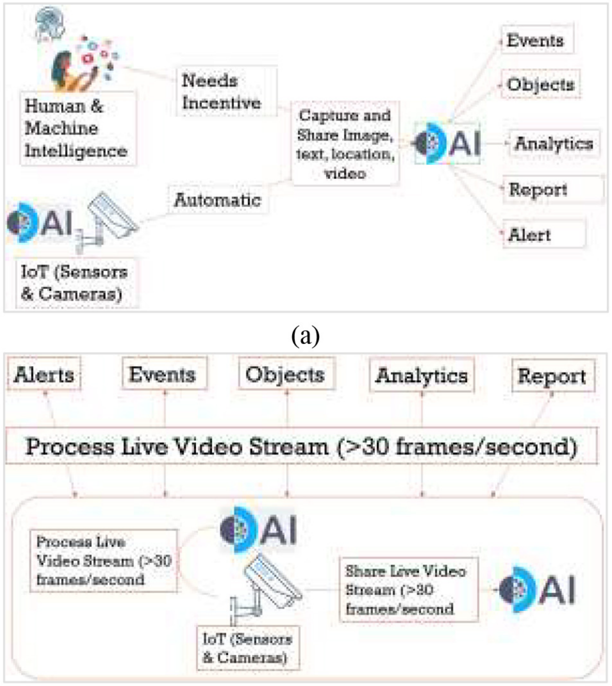
Ahmed Ghoneim is with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia, and also with the Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebeen El-Kom 32511, Egypt (e-mail: ghoneim@ksu.edu.sa).

Digital Object Identifier 10.1109/JIOT.2021.3109435

I. Introduction

INDUSTRIAL Internet of Things (IIoT) has contributed to the recent smart city development through real-time sensing capabilities [1]. Urban city management and automation has heavily relied on the IIoT [2]. For example, IIoT has helped automating tasks at the constrained edge nodes [3] in healthcare applications [4], [24] in a smart city. Several core technological advancements have contributed to this growth. The memory, hardware, operating system, and communication modules that used to be constrained on IIoT devices in the past are no longer a constraint. State-of-the-art edge IIoT devices started becoming resources, such as a laptop or a PC. Massive volumes of IIoT data can now be used to train artificial intelligence (AI) applications to accurately predict different events and phenomena [5]. Recent IIoT devices can run Linux OS with large memory and a 5G transceiver module for ultra-speed communication with the outside world [6]. Hardware vendors, such as Intel, Nvidia, and Google provide GPU capabilities to edge devices so that deep learning applications can be trained and deployed on edge devices [7]. The deep learning model providers, such as Facebook and Google started building models that can be ported to the edge IIoT devices or smartphones.

To add to the existing advancements, diversified types of faster object detection and recognition deep learning libraries have been made available by opensource communities. State-of-the-art deep learning models can track hundreds of objects and events from real-time cameras and IIoT data streams, which could only be performed by powerful cloud computers in the past. These deep learning models can now be deployed to drones, cars, edge devices, buildings, street lights, cameras, and so on [8]. This has opened a new horizon of tracking smart city events such as fire, flood, accident, violence, and so on at the edge in real time. Thanks to the faster AI processes, the processed events can now be reported by humans or the distributed IIoT edge devices with evidence. While the Industry 4.0 was mainly focused on the advancements of the industries, a new initiative has begun [9] where a human itself is in the center of the IIoT ecosystem. While Industry 4.0 was more focused toward urbanization and IIoT advancements, the human aspects within the smart city was not mandated. Industry 4.0 was focused toward best quantity and IIoT-based automated mass production—missing the smart life standard. Hence, a paradigm shift is happening toward Industry 5.0 in which IIoT is focused toward human-centric computing. This will allow humans to be in charge of IIoT ecosystem [10].


Human & Machine Intelligence
Needs Incentive
Automatic
IoT (Sensors & Cameras)
Capture and Share Image, text, location, video
AI
Events
Objects
Analytics
Report
Alert
(a)
Alerts
Events
Objects
Analytics
Report
Process Live Video Stream (>30 frames/second)
Process Live Video Stream (>30 frames/second
Share Live Video Stream (>30 frames/second
IoT (Sensors & Cameras)

(b)

Fig. 1. Different scenarios of AI-enabled IIoT processes (a) AI at the cloud and (b) processing IIoT data at the edge at real time.

Fig. 1 shows different scenarios of smart city event detection. Fig. 1(a) presents the scenario where the IIoT sensors and surveillance cameras, whether static or mobile, stream the content to a cloud AI machine, where scenes within the stream are being classified by several AI applications to deduce diversified types of objects and events. A city authority can use the outputs of AI algorithms to act accordingly. Fig. 1(a) uses AI capability to infer the events and objects at the edge and share the inferencing results with the authority. Fig. 1(a) shows another dimension where specialized smartphone apps can be provided to city residents. With appropriate incentive, city residents can report different events and objects with evidence, through their human intelligence. In addition to human intelligence, the IIoT edge devices having AI at the edge can also report similar events and objects. Finally, Fig. 1(b) shows the recent development in faster object detection and recognition models such as You only Look once (YoLo)V5, which can detect objects on live video streams with more than 25 frames/s. YoLoV5 shows superior object detection within real-time video streams in terms of higher frames/second. YoLoV5 can be pipelined with other algorithms such as DeepSort, where objects within video streams can be sorted and tracked separately. The speed of object detection and tracking quality is achieved with a slight decrease in accuracy in the case where low-level details are needed such as skeletal tracking of humans in real-time.

Coupled with 5G low-latency communication capability and faster AI processes on live IIoT data, a smart city can collect rich city events for city management. Smart cities around the world started to leverage such advancements. For example, the city of London, U.K., has made some city cameras in more than 1000 locations public for handling such city event monitoring. Combining the AI-based automated camera objects and events detection capability at any given time would give the city authority the high-level of what is happening in a road, within a geofence or within an area of interest. Moreover, multiple cameras having common physical surveillance regions can be configured in a peer-to-peer network where mutually exclusive geo-zones are monitored by each camera to avoid colluding event detection with each other or providing temporally connected event detection from multiple cameras.

Adding human intelligence with IIoT data and AI capability, diversified types of events and objects can be recognized, and appropriate analytics, reports, and alerts can be generated. These events can then be sorted out by automated AI algorithms, evidence in terms of text, image, audio, video, and location can be attached to the reported event, and finally used as a ticket that can be managed by the respective responsible departments in a smart city.

Deep learning applications [28] have been getting much attention from the research community and as a result, we observe models that have very high accuracy in processing real-time IIoT data. For example, IIoT interfaced with ac unit within an industry can be controlled based on AI. One scenario is an IIoT camera observes the human presence within a facility and if no human is detected for a duration, the ac unit can be shut down by the telemetry signal. Intelligent perimeter protection, intelligent detection of objects from satellite images, and in many such scenarios, the IIoT can be controlled from the AI algorithms. While much work on the IIoT has happened in the past, the challenge remains in handling the IIoT data in real time due to the constraint of edge inferencing. Toward the automated smart city event management through AI-enable IIoT, in this article, we propose the following contributions.

1) We use a combination of convolution neural network (CNN) and recurrent neural network (RNN) to deliver object and event detection and recognition from live video cameras and IIoT data having temporal dimension.
2) We have tested our framework's object and high-level event surveillance accuracy with 1000 Transport for London (tfl) cameras in London, U.K. and 500 YouTube live cameras around the globe.
3) Incorporating YoLoV5 faster deep learning model, our custom-trained event and object detection and recognition models can track objects and events in live cameras.
4) We have developed a dashboard for smart city authority that shows the aggregated AI output for each high-level event context.
5) Using our proposed framework, IIoT devices' operations can be controlled manually by the city authority based on the AI inferencing results or AI logic can be given authorization by city authority to act on its own logic.

The remainder of this article is organized as follows. In Section II, we have presented some state-of-the-art related works. In Section III we have illustrated the design of the proposed system. In Section IV, we have detailed the implementation and test results. Finally, we conclude the article with future works in Section V.

II. Related Works

Liu *et al.* [11] have proposed a deep reinforcement learning framework for sharing IIoT data in a secure manner. The data shared during federated deep learning in IIoT pose privacy threats [29]. Hence, Zhang *et al.* [12] have studied and proposed a privacy protection mechanism of IIoT data in the context of deep learning. In another effort, blockchain and federated learning has been proposed by Lu *et al.* [13] to offer privacy of IIoT data. Federated deep learning has been used by Yin *et al.* [14] to establish IoT device-to-device collaboration among IIoT. In order to provide trustworthiness and data privacy in the IIoT network, the framework named PriModChain [15] incorporates differential privacy, federated learning, and blockchain. Building trust in the cyber–physical system (CPS), where IIoT nodes play critical roles is an essential element. Lv *et al.* [16] have proposed a trustworthy model for accessing CPS. Deep learning has been used in IIoT devices for edge surveillance. Zhao *et al.* [7] have claimed 16 frames/s object detection speed with 89% precision. Transfer learning-enabled blockchain has been used to provide secure authentication of IIoT data [17]. Training at the IIoT edge has been explored by Liang *et al.* [8] in which the authors propose a lightweight deep learning algorithm that can be used to train and also run the model at the edge IIoT device without compromising the quality of the inferencing.

Advances in IIoT communication speed is noteworthy. Kato *et al.* [18] envisioned the 6G features that will be offered to IIoT-based applications. Deep learning has been studied by researchers [1] in the context of IIoT data security and scalability. A 5G aided IIoT architecture is proposed in [6], which leverages transfer learning [19]. Data communication among AI components within 6G network will pose security vulnerabilities. Hence, Li *et al.* [20] have proposed a blockchain-based provenance framework for such networks. 5G-enabled IoT in the context of Internet connected vehicles has been studied in [21]. The usage of fog computing near the edge has also been studied by Yassine *et al.* [22] and Hossain and Muhammad [23] IoT data processing. Leveraging beyond 5G network, the work presented in uses deep learning at the edge to avoid IIoT data privacy leaks.

Detecting anomalous events from live IoT camera feed is an interesting area of research. Ahmad and Conci [25] have used deep learning techniques to detect events from multimedia IoT sensors. Complex event detection from camera feed using event calculus has been proposed in [26]. Although past research works have addressed deep learning applications in the context of IIoT, very few works have addressed the usage of deep learning on live IIoT data such as surveillance cameras. In this article, we strive to contribute to the area of deep learning at the edge IIoT with live stream.

III. System Design

A. System Architecture

1) IIoT: The IIoT devices consist of industrial grade IoT devices that can perform diversified types of event detection in real time. To make it deployable at the city level and add deep learning capability at the edge, we have chosen host environment as raspberry PI, Nvidia Jetson Nano, and Nvidia Jetson Xavier AGX. While the Nvidia family of devices come with GPU, the raspberry PI was augmented with either Intel NCS2 or Google CORAL TPU. As for the smartphone, we have added a FLIR thermal camera. Additional sensors via Arduino are done through GPIO pins of the raspberry PI or Nvidia boards. Fig. 2 shows some of the IIoT devices that are used in this research.

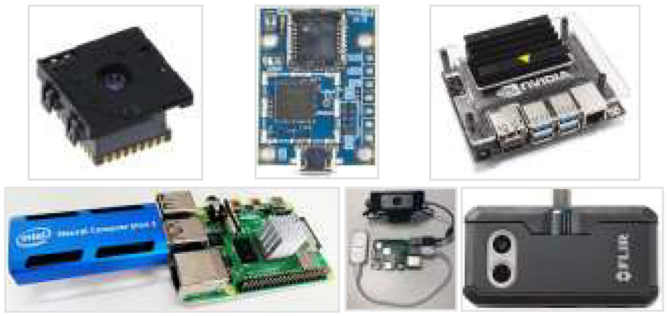


Fig. 2. Some of IIoT devices that are used in this research.

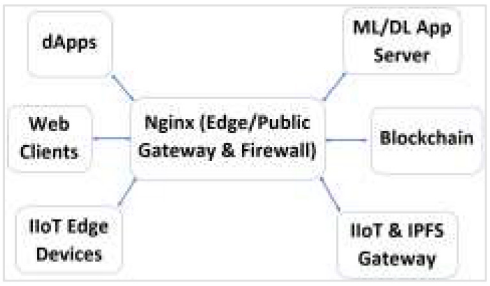


Fig. 3. Cloud and edge architecture to support AI-enabled multimedia IIoT data processing.

2) Edge and Cloud: Our proposed edge and cloud environment is shown in Fig. 3. The edge environment consists of Raspbian OS for python 3.8 and a virtual environment installed. The Nvidia family of boards have been configured with Ubuntu 18.04 LTS. The smartphone devices have also been considered as edge device. IIoT devices have been configured with MQTT protocol to share the IIoT data via the MQTT gateway to our local cloud. The local cloud has been configured with NginX, which is used as a Web server, reverse proxy, and load balancer. The private cloud hosts the AI models. Although all the AI models work individually and independently, a controller, such as the case of a Model-View-Controller pattern, manages the AI task to be performed, the appropriate AI models to be invoked, and once the results are available, appropriate view or further IIoT control signal generating models are invoked. The AI models are distributed into multiple clusters of the GPUs and each AI algorithm is programmed to consume a certain amount of GPU. The controller forwards the AI task to all the AI modules running in parallel threads.

3) Overall Framework: The overall framework glues together multiple entities to perform smart city monitoring tasks. The sensors and controller layer host the physical IIoT sensors that perform different sensing tasks. These IIoT sensors are interfaced with the edge computing and communication modules. We have tested with 5G hardware attached on top (HAT) as an expansion board on top of an edge node. As

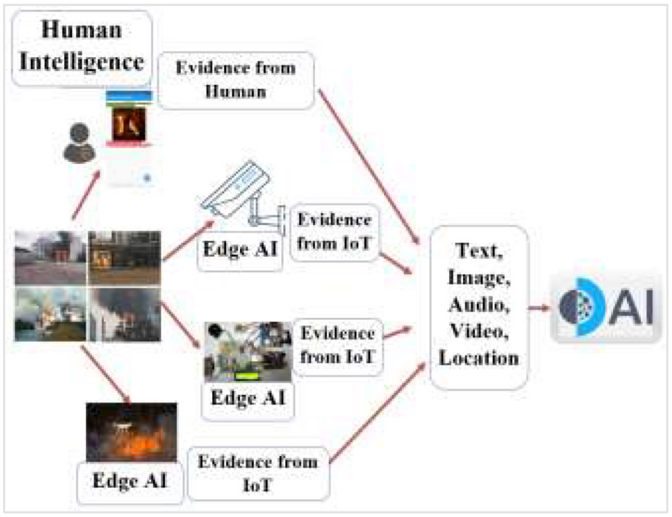


Fig. 4. Combining the outputs of individual AI tools at the edge and share with city cloud.

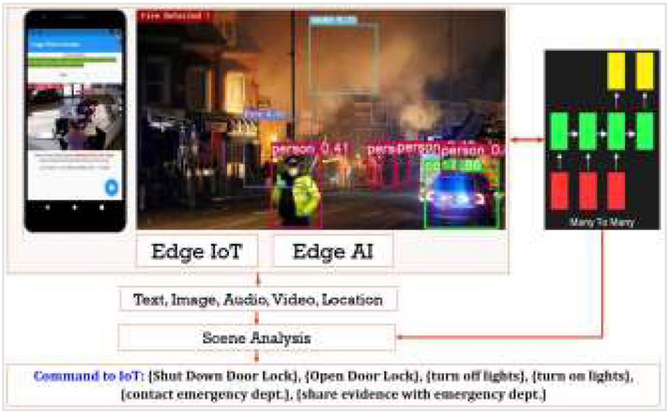

person 0.41
person
Many To Many
Edge IoT
Edge AI
Text, Image, Audio, Video, Location
Scene Analysis
Command to IoT: {Shut Down Door Lock}, {Open Door Lock}, {turn off lights}, {turn on lights}, {contact emergency dept.}, {share evidence with emergency dept.}

Fig. 5. Deducing multiple outputs from the multiple users and IIoT inputs within feedback loop.

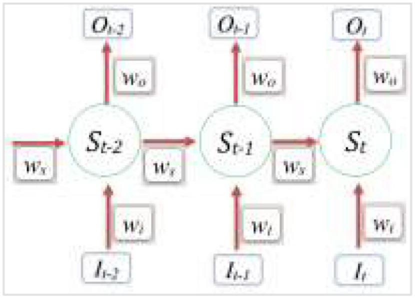


Fig. 6. Neural network with memory to remember past events and classify the current events and objects.

mentioned in Section III-A1, we have developed different lite AI models targeting different edge devices. This edge learning environment allows either the edge model to be trained locally at the IIoT device edge or running inferencing on the local data or events at the edge. To make the IIoT data and the inferencing results immutable, we use both interplanetary file system (IPFS) and private blockchain. Blockchain also establishes trust and provenance among IIoT nodes and human users that relate to the IIoT ecosystem. Finally, several smart city event reporting and services have been developed as a proof concept, and the details of these applications are described in Section IV.

4) Aggregation of Event and Objects From IIoT: When an event, accident, or disaster happens in a city, many images, audio, video, text, and location data are being generated by either humans or IIoT devices. These multiple media analyses give a rich amount of information to the smart city authority so that appropriate departments can handle the event(s). Assuming we have N number of AI algorithms that are trained to detect, recognize, and predict N number of events from M number of media sources, combining N number of deduced results would produce a rich scene analysis in real time. Fig. 4 shows how we deduce multiple outputs from the same input video stream and aggregate the individual inference results. Individual edge nodes employ multiple AI models that are deployed locally and perform inferencing tasks simultaneously. For example, during a fire event, the fire detection model detects fire and smoke, the human presence detection model detects the human presence, and the emergency vehicle detection model detects the emergency vehicles from the same camera stream. The aggregated results are used for further analysis.

5) AI to Control IIoT: Fig. 5 shows another dimension where events, objects, reports, and alerts are generated from the live video scene analysis and control signals are generated for IIoT devices. The optical flow field theory has been used to capture the velocity of the moving objects within the camera stream. The optical flow allows tracking the instantaneous velocity and direction of each moving pixel among adjacent video frames. Aggregating the spatially connected pixels of the tracked objects provides us the tracked event. In this scenario, a CPS is controlled by AI, which is sometimes called the Autopilot mode, in which the deep learning model is empowered with taking decisions based on the inferencing results. An example scenario is using many-to-many RNN [27]. Since the autopilot mode allows the access and control of physical world entities through AI decision, we have developed a zero-trust system that incorporates biometric authentication, multifactor authentication, push notification, and digital certificates in different scenarios of IIoT control.

B. System Modeling

In our proposed deep neural network model, we train a network at a specific time, teaching the network all what happened before so that the temporal changes among the IIoT data are stored. Fig. 6 shows a high-level neural network that memorized past states.

The state values are calculated as follows [27]:

$$S_t = \varphi(I_t * w_i + s_{t-1} * w_s) \tag{1}$$

where

φ = "relu",
w_i = input weight
w_o = output weight
w_s = weight of each state.

Output at stage t

$$O_t = \{(S_t * w_o)|\text{SoftMax}(S_t * w_o)\}. \tag{2}$$

RNN compares the current output O_t with desired output O_d to find the error E_t. Hence, error at stage t

$$E_t = (O_d - O_t)^2 \tag{3}$$

where O_d = desired output and O_t = current output at the last stage. To update each weight matrix, we need to subtract the partial derivative of each weight w.r.t their errors. Gradient descent can be used to update the weight. From (2) and (3), we calculate the partial derivative of the error w.r.t the output weight as follows:

$$\frac{\partial E_t}{\partial w_o} = \frac{\partial E_t}{\partial O_o} * \frac{\partial O_t}{\partial w_o}. \tag{4}$$

We can expand the backpropagation by calculating the error w.r.t the state weight as follows:

$$\frac{\partial E_t}{\partial w_s} = \frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial w_s} \tag{5}$$

$$\begin{aligned}\frac{\partial E_t}{\partial w_s} &= \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial w_s}\right)\\ &\quad + \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial S_{t-1}} * \frac{\partial S_{t-1}}{\partial w_s}\right)\end{aligned} \tag{6}$$

$$\begin{aligned}\frac{\partial E_t}{\partial w_s} &= \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial w_s}\right)\\ &\quad + \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial S_{t-1}} * \frac{\partial S_{t-1}}{\partial w_s}\right)\\ &\quad + \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial S_{t-1}} * \frac{\partial S_{t-1}}{\partial S_{t-2}} * \frac{\partial S_{t-2}}{\partial w_s}\right).\end{aligned} \tag{7}$$

The gradient descent w.r.t to the input weight w_i can be calculated as follows:

$$\begin{aligned}\frac{\partial E_t}{\partial w_i} &= \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial w_i}\right)\\ &\quad + \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial S_{t-1}} * \frac{\partial S_{t-1}}{\partial w_i}\right)\\ &\quad + \left(\frac{\partial E_t}{\partial O_t} * \frac{\partial O_t}{\partial S_t} * \frac{\partial S_t}{\partial S_{t-1}} * \frac{\partial S_{t-1}}{\partial S_{t-2}} * \frac{\partial S_{t-2}}{\partial w_i}\right).\end{aligned} \tag{8}$$

As for training, we use a combination of CNN and RNN since we work on both live video cameras and IIoT data. In the case where the live camera module is used, the CNN first reduces the dimension of 2-D images $x(t)$ into 1-D vector $z(t)$ by

$$f_{\text{CNN}}(x(t)) = z(t). \tag{9}$$

The RNN decoder then receives the input vector $z(t)$ and outputs categorical predictions on the input live video or IIoT data stream.

IV. Implementation and Test Results

For the Proof of Concept (PoC), we have developed the following[1] smart city crowdsourcing as well as their automated IIoT enabled AI systems.[2] In both scenarios, AI has been used both on edge IIoT devices and at the cloud. We have also developed a citizen smartphone app in flutter (see Fig. 5) that will allow humans to report events to a city authority. The smartphone also has the similar number of services as PoC. Using the smartphone app, city residents can report road accident, flood, road construction, path hole or blockage, violence, unauthorized drone movement, car number plates and human injury involved in accidents, fire and smoke, trash bin to be cleaned, road under water, and so on. Similar events can be captured by the AI-enabled IIoT devices. We have developed individual AI modules and an integrator of all these AI modules such that once an image or video is uploaded by either city residents or IIoT sensor, it is being classified by respective AI algorithms to deduce multiple scenarios from the live camera streams. Multiple algorithms produce rich amount of information from the scene analysis. Once all these individual AI components' reports are being combined, it is shared with the city authority for further actions. As for the AI-enabled control of IIoT devices, we have developed a scenario, where an IIoT camera observes human activity within a region of interest and can turn on/off the ac, open/close the door lock, and send an SMS with the camera scene analysis result. The IIoT can be controlled by the AI algorithm in this mode. Next, we will explain different implemented modules and corresponding test results we have obtained.

As for the RNN algorithm, we have used long short-term memory (LSTM) network and for CNN, we have used several base Object Detection Deep Learning models, such as ResNet-152, MobileNetV2, InceptionV3, and InceptionResNetV2 through Keras API. We have used both Keras and Pytorch for training the models. Nvidia GeForce RTX 2070, RTX 2080, RTX 2080 Ti, and RTX 3090 have been used for training the CNN-based models. Flutter with dart programming language has been used for the smartphone and Web application development. The smart city authority dashboard has been designed with Flask.

A. Preparation of Data Sets

We have taken three different approaches of preparing the training, validation, and test data sets. First, we have tried to search the data set in Kaggle and other opensource repositories. Most of the data sources are already available in Kaggle. Second, because of the nature of this research, we had to build our own data sets. In that case, we have downloaded images from Google and manually labeled them using opensource labelImg[3] and saved the annotations in either the PascalVOC or YOLO format, depending on the targeted model. Examples of such cases are flipped car, trash bins, and other types of smart city incidents, where we used YoLoV5 for live object detection. Finally, some of the events that have a temporal relationship within the data set, e.g., violence detection requires the video data set. In this case, we have managed YouTube video clips containing the right type of events and scenes.

B. Edge IIoT

Since the IIoT devices have constrained computing, memory, and deep learning capabilities, we have developed

[1] https://ai.upm-research.net/

[2] https://ai.upm-research.net/dashboard/

[3] https://github.com/tzutalin/labelImg

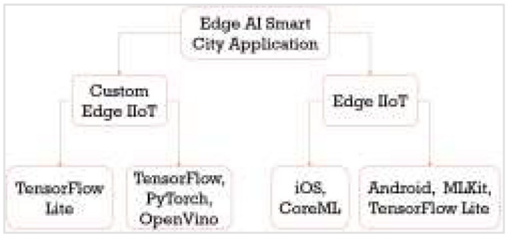

Edge AI Smart City Application
Custom Edge IIoT
Edge IIoT
TensorFlow Lite
TensorFlow, PyTorch, OpenVino
iOS, CoreML
Android, MLKit, TensorFlow Lite

Fig. 7. Classification on edge AI models.

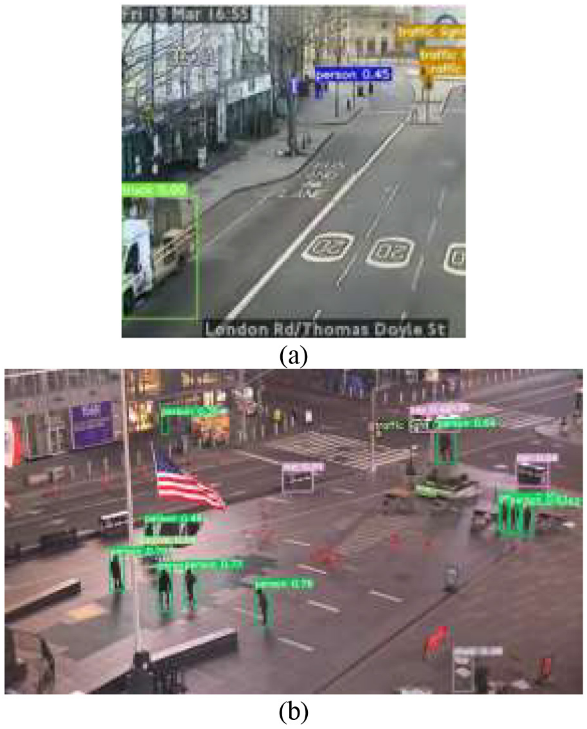


Fig. 8. Recognized multiobjects from (a) London tfl camera streams and (b) YouTube live streams.

edge models depending on the type of IIoT devices. Fig. 7 shows several edge environments that we have tested in this research. In all these cases, we have converted the full-sized model into corresponding lite version that can run on the IIoT devices. Once the IIoT devices are loaded with the respective deep learning model, diversified types of event detection and prediction has been performed.

C. Deducing Event, Objects, Alert, and Analytics From AI-Enabled Live IIoT

To test the performance of the developed real-time analytics, we have subscribed to 1000 live cameras in the city of London from the tfl[4] API. We have developed a python backend that connects to the live stream based on the street, or (latitude, longitude, radius) and passes the stream via our aggregated deep learning application. Fig. 8(a) shows a sample output from one of the live cameras in which our aggregated object and event detection algorithm detects live events and objects of interest. Fig. 8(b) shows another dimension in which YouTube live 4K video channels around the globe are fed to our pafy[5] streamer backend, where custom trained yolov5[6] deduces the classification results with bounding boxes around 25 frames/second.

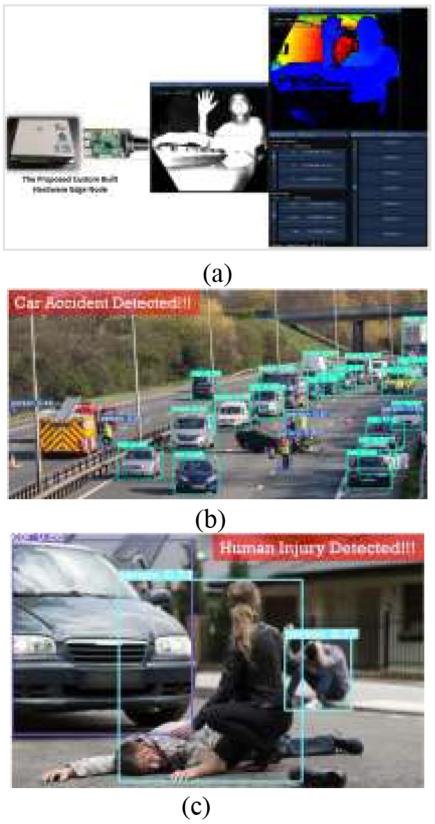


Fig. 9. Deducing multiple events from IIoT data frames (a) audio, thermal camera for deducing human emotions, (b) observing multiple objects within an accident scenario, and (c) low-level details of the accident scene deduced from IIoT.

In the case of deep learning applications, sometimes one video stream contains multiple types of incidents. As shown in Fig. 9, each single image taken from a video stream shows more than one type of recognized objects and events using the similar optical flow algorithm as described in Section III-A5. For example, Fig. 9(b) shows an accident scene with a number of vehicles on the scene, human counts, specialized bounding box around the vehicle(s) under accident, vehicle is under fire or not, etc., can be deduced by combining results from aggregated IIoT sensory data. Similar rich results can be obtained from the scenarios shown in Fig. 9(a) and (c). When the proposed system detects images and identifies accidents from IIoT data, this can be linked to the emergency systems to send ambulance to the scene as quick as possible. This will reduce the risk for the individuals and ensure fast response to the accident scene, thereby saving lives.

D. AI-Enabled IIoT

In this research, we have developed several proof-of-concept IIoT infrastructure in which IIoT devices can be controlled by the AI-deduced logic. In this mode, each IIoT device can be either manually controlled by a human actor having three

[4]https://www.tfljamcams.net/

[5]https://pypi.org/project/pafy/

[6]https://github.com/ultralytics/yolov5

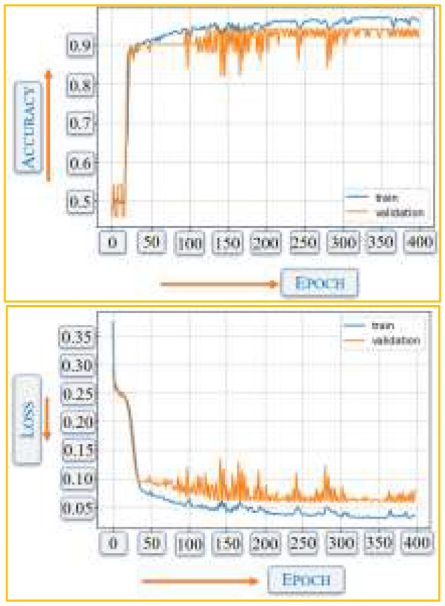


Fig. 10. Training and validation accuracy and loss for the proposed module.

layers of security or by human authorized an AI algorithm. In both cases, the human actor must go through a biometric checking followed by a multifactor authentication mechanism and a final system login. A human actor activates this mode so that the IIoT devices can be controlled or monitored by the AI algorithms. In this mode, AI is trained to deduce the objects and events from IIoT and based on the scenario, appropriate IIoT modules are activated.

E. Model Training and Performance Measurement

We have followed performance benchmarking measures to find out the best model and training environment. For example, we have observed the effects of adding more deep neural network layers on the speed of the training process. As we increase the number of layers, the training speed decreases. To compensate the speed cap, we have upgraded the GPU memory. The main motivation of increasing the number of layers is to check whether the train/validation/test accuracy of the trained model increases. Different deep learning models exhibit different patterns of the optimum number of layers to achieve the highest accuracy. Hence, we have accommodated these benchmarks while choosing the right training parameters and environment.

The violence detection module has been trained with the RWF-2000 video data set. The training video data set contains 2000 clips, each clip is of 5-s duration and 30 frames/s. Each clip is labeled as violent or nonviolent that are captured from live surveillance cameras. Fig. 10 shows the training and validation accuracy and loss, respectively. Object detections and recognition and scene analysis with bounding boxes from live camera feed generally requires specialized algorithms, such as Faster-RCNN, Mask-RCNN, SSD, and YoLo. The accuracy and loss values for both validation and training process closely follow each other without much deviation. The accuracy level achieved by the final model is above 96% while loss remains below 10%. Classification loss w.r.t the bounding

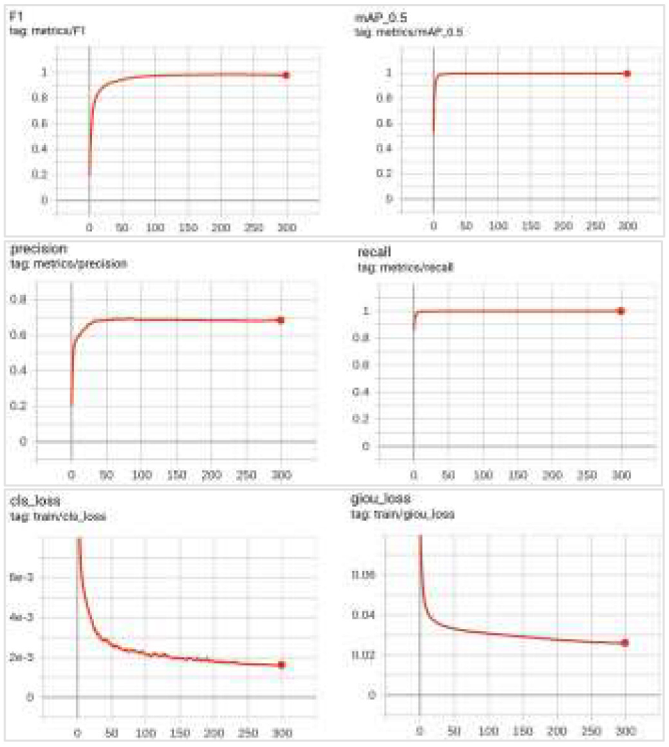


Fig. 11. Performance metrics of the proposed module.

TABLE I
PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH INDIVIDUAL MODELS AT THE SERVER GPU MACHINE

	Accuracy	**precision**	**recall**	**F1-score**	**QWK score**
Proposed Model	*0.9778*	*0.9780*	*0.9778*	*0.9777*	*0.9888*
EfficientNet	0.9715	0.9738	0.9746	0.9724	0.9766
VGG-16	0.9629	0.9713	0.9732	0.9681	0.9790
ResNet50	0.9753	0.9756	0.9753	0.9752	0.9731
DenseNet201	0.9425	0.9428	0.9425	0.9420	0.9591
InceptionV3	0.9360	0.9361	0.9360	0.9356	0.9590
InceptionResNetV2	0.9565	0.9573	0.9565	0.9558	0.9779
Xception	0.9565	0.9570	0.9565	0.9559	0.9775

boxes is measured through cls_loss, which is a cross-entropy loss. Hence, metrics, such as mean average precision (mAP), Intersection over Union (IoU), and bounding box classification loss have been used. Fig. 11 shows a tensorboard log analysis in terms of the $F1$ score, mAP at IoU 0.5 as threshold, precision, recall, cls_loss, and general IoU (gIoU) loss values. As reported by the similar research, the F1 score, mAP, precision, recall, cross-entropy loss, and gIoU values seem to be within the acceptable range.

We have setup a comparison metrics as shown in Tables I and II in which our proposed model ensembles average of seven deep learning models, namely, EfficientNet, VGG-16, ResNet50, DenseNet201, InceptionV3, InceptionResNetV2, and Xception. We have then compared the results obtained in two scenarios. Table I shows the performance comparison of individual models while Table II shows the performance comparison with respect to the ensemble minus one model. The ensemble model has total params: 167 190 854, trainable params: 84 030 398, and nontrainable params: 83 160 456. As shown in

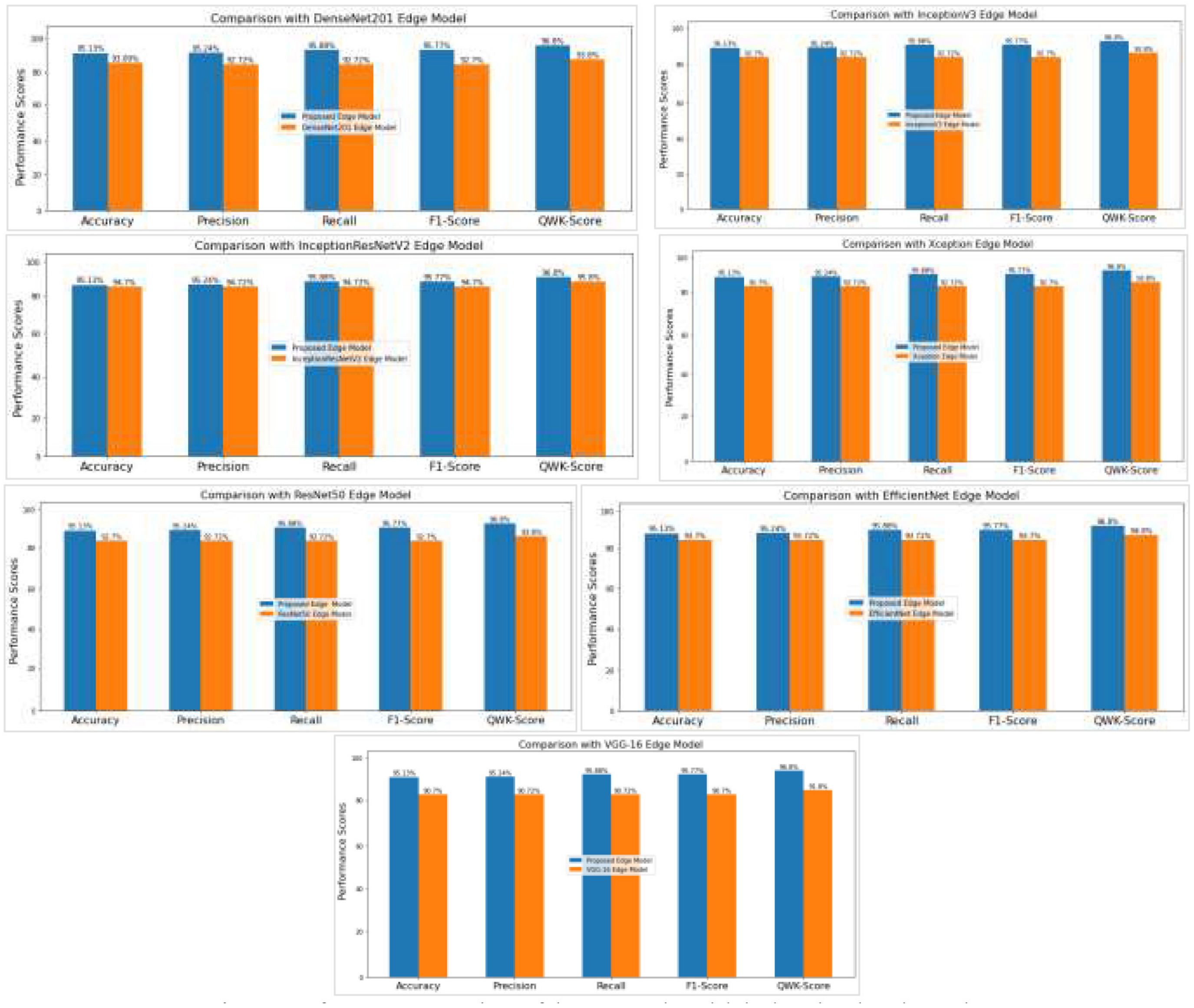


Fig. 12. Performance comparison of the proposed model deployed at the edge node.

Tables I and II, our proposed method outperforms the compared models. These models are compared at the server-side GPUs running the algorithms. Among the compared individual models, as Table I shows, ResNet50 shows the closest performance while Ensemble model-Except InceptionResnetV2 shown in Table II exhibits the closest performance in the ensemble group. Among the compared models, our proposed model has accuracy, precision, recall, F1-score, and QWK-score has achieved slightly better than the rest of the models. This signifies that the fusion of seven models have produced better performance in all categories. The most interesting results have been found to be in Table II. Fusing more models seems to produce better performance results in comparison to the regular models shown in Table I. Almost all the performance measures have increased in Table II when compared to Table I models. However, in both cases, our proposed model has shown slightly better performance.

Fig. 12 shows the performance comparison of the edge models. As shown in Figs. 2 and 7, we have ported our proposed algorithm in the edge devices. All the full models shown in Tables I and II have been converted to an edge model, e.g., TFLite and then compared with the violence detection from live camera feed scenario. As shown in Fig. 12, our proposed model outperforms all the compared models. However, the performance gain came with a price of a greater number of trainable parameters and more memory requirement at the edge node. We believe the future edge nodes will support more storage space and this shortcoming will not be a constraint.

In this article, we have tried to test a set of deep learning applications running on the edge IoT devices. The edge devices capture different events from video streams using either edge model or a cloud model. We have tested the algorithms' and models' performance through eight instances of applications that are related to smart city monitoring. The target was to see whether the real-time or live events can be captures and reported via edge learning. The test results show that the framework has potential to be used for smart city applications. However, there are several challenges that

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH ENSEMBLE MODELS AT THE SERVER GPU MACHINE

	Accuracy	**precision**	**recall**	**F1-score**	**QWK Score**
Proposed Model	*0.9778*	*0.9780*	*0.9778*	*0.9777*	*0.9888*
Ensemble Model - Except ResNet50	0.9667	0.9668	0.9667	0.9664	0.9822
Ensemble model Except DenseNet201	0.9667	0.9669	0.9667	0.9664	0.9822
Ensemble model - Except VGG16	0.9645	0.9647	0.9645	0.9641	0.9801
Ensemble model - Except InceptionNet	0.9669	0.9671	0.9647	0.9586	0.9819
Ensemble model - Except InceptionResnetV2	0.9689	0.9691	0.9689	0.9686	0.9827
Ensemble model - Except XceptionNet	0.9686	0.9688	0.9686	0.9685	0.9884

we have faced. First, the heavy GPU demand at the cloud for the applications running as tenant under the same physical machine. Multithreading and GPU memory allocation per application needed to be done. Edge GPU is also a constraint for the edge learning processes. We hope these hardware limitations will not be a barrier in the coming days.

V. CONCLUSION AND FUTURE WORKS

In this article, we have investigated the usage of AI-enabled IIoT data for smart city event reporting. We have developed several deep learning applications that can run on both edge IIoT devices and in the cloud. Using our framework, a smart city authority can collect live events from city residents and deployed IIoT devices so that the events can be handled by the responsible departments. Our developed AI-enabled IIoT can report events autonomously. Particularly, we have tested scenarios where deep learning applications can work on live cameras, detect, and recognize events and objects and deduce high-level scenes. We have also proposed the autopilot mode in which IIoT devices can be guided and controlled by the AI logic. This will allow the city to automate many event detections to reporting and controlling the ecosystem. As for future works, we started reaching out to several city councils so that our proposed solutions can be deployed.

REFERENCES

[1] Y. Wu, H.-N. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2300–2317, Feb. 2021.

[2] T. Bányai *et al.*, "Optimization of municipal waste collection routing: Impact of industry 4.0 technologies on environmental awareness and sustainability," *Int. J. Environ. Res. Public Health*, vol. 16, no. 4, p. 634, 2019.

[3] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: Performance of resource-constrained IoT networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 174–183, Jan. 2021.

[4] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT)—Enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, Jun. 2016, doi: 10.1016/j.procs.2019.09.104

[5] A. Fu, X. Zhang, N. Xiong, Y. Gao, and H. Wang, "VFL: A verifiable federated learning with privacy-preserving for big data in Industrial IoT," 2020. [Online]. Available: arXiv:2007.13585.

[6] K. Wang *et al.*, "Voice-transfer attacking on industrial voice control systems in 5G-aided IIoT domain," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7085–7092, Oct. 2021.

[7] Y. Zhao, Y. Yin, and G. Gui, "Lightweight deep learning based intelligent edge surveillance techniques," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1146–1154, Dec. 2020.

[8] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward edge-based deep learning in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4329–4341, May 2020.

[9] A. Haleem and M. Javaid, "Industry 5.0 and its applications in orthopaedics," *J. Clin. Orthop. Trauma*, vol. 10, no. 4, pp. 807–808, 2019.

[10] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot co-working," *Procedia Comput. Sci.*, vol. 158, pp. 688–695, Jun. 2019. [Online]. Available: https://doi.org/10.1016/j.procs.2019.09.104

[11] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for Industrial IoT with deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3516–3526, Jun. 2019.

[12] X. Zhang, X. Chen, J. K. Liu, and Y. Xiang, "DeepPAR and DeepDPA: Privacy preserving and asynchronous deep learning for Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2081–2090, Mar. 2020.

[13] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[14] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6348–6359, Jul. 2020.

[15] P. C. M. Arachchige, I. Khalil, P. Bertok, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in Industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6092–6102, Sep. 2020.

[16] Z. Lv, Y. Han, A. K. Singh, G. Manogaran, and H. Lv, "Trustworthiness in Industrial IoT systems based on artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1496–1504, Feb. 2021, doi: 10.1109/TII.2020.2994747.

[17] X. Wang, S. Garg, H. Lin, M. J. Piran, J. Hu, and M. S. Hossain, "Enabling secure authentication in Industrial IoT with transfer learning empowered blockchain," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7725–7733, Nov. 2021.

[18] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.

[19] P. K. Deb, A. Mukherjee, T. Sarkar, and S. Misra, "Magnum: A distributed framework for enabling transfer learning in B5G-enabled Industrial-IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7133–7140, Oct. 2021.

[20] W. Li, Z. Su, R. Li, K. Zhang, and Y. Wang, "Blockchain-based data security for artificial intelligence applications in 6G networks," *IEEE Netw.*, vol. 34, no. 6, pp. 31–37, Nov./Dec. 2020.

[21] M. A. Rahman, M. S. Hossain, M. M. Rashid, S. Barnes, and E. Hassanain, "IoEV-Chain: A 5G-based secure inter-connected mobility framework for the Internet of Electric Vehicles," *IEEE Netw.*, vol. 34, no. 5, pp. 190–197, Sep./Oct. 2020.

[22] A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, "IoT big data analytics for smart homes with fog and cloud computing," *Future Gener. Comput. Syst.*, vol. 91, pp. 563–573, Feb. 2019.

[23] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5G," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2399–2406, Aug. 2018.

[24] M. A. Rahman, M. S. Hossain, N. A. Alrajeh, and F. Alsolami, "Adversarial examples—Security threats to COVID-19 deep learning systems in medical IoT devices," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9603–9610, Jun. 2021.

[25] K. Ahmad and N. Conci, "How deep features have improved event recognition in multimedia: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 2, pp. 1–27, 2019, doi: 10.1145/3306240.

[26] A. Khan, L. Serafini, L. Bozzato, and B. Lazzerini, "Event detection from video using answer set programming," in *Proc. CEUR Workshop*, vol. 2396, 2019, pp. 48–58. [Online]. Available: http://ceur-ws.org/Vol-2396/paper25.pdf

[27] K. Munir, H. Elahi, A. Ayub, F. Frezza, and A. Rizzi, "Cancer diagnosis using deep learning: A bibliographic review," *Cancers*, vol. 11, no. 9, p. 1235, 2019.

[28] M. S. Hossain and G. Muhammad, "Deep learning based pathology detection for smart connected healthcare," *IEEE Netw.*, vol. 34, no. 6, pp. 120–125, Nov./Dec. 2020.

[29] Y. Liu *et al.*, "Deep anomaly detection for time-series data in Industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.