IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 January 2024, accepted 7 February 2024, date of publication 16 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3366693

RESEARCH ARTICLE

PublicVision: A Secure Smart Surveillance System for Crowd Behavior Recognition

MARWA QARAQE[1], (Senior Member, IEEE), ALMIQDAD ELZEIN[1], EMRAH BASARAN[1], YIN YANG[1], (Member, IEEE), ELIZABETH B. VARGHESE[1], WISAM COSTANDI[2], JACK RIZK[2], AND NASIM ALAM[2]

[1]Division of Information and Computing Technology, College of Science and Engineering, Hamad bin Khalifa University, Qatar Foundation, Doha, Qatar
[2]Informatica Qatar (IQ), Doha, Qatar

Corresponding author: Elizabeth B. Varghese (evarghese@hbku.edu.qa)

This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) under Grant AICC03-0324-200005 and by the Qatar National Library (Open Access Funding).

ABSTRACT Crowd behavior recognition plays a critical role in various domains, including public safety, event management, and urban planning. Understanding crowd dynamics and detecting behaviors based on violence levels are crucial for preventing incidents and maintaining order in crowded environments. However, traditional surveillance methods fall short of providing comprehensive and real-time insights into complex crowd behavior patterns and fail to distinguish different violence levels within crowds that affect proactive decision-making. Moreover, most of the current systems do not provide reliable secure data transmission and are not viable in protecting the privacy of individuals. This paper designs an end-to-end secure and smart surveillance system, namely **PublicVision**, that transmits CCTV data securely to a remote central hub where a deep learning (DL) model based on Swin Transformer is utilized to identify and analyze crowd behaviors. A novel video dataset was created to train the DL model that identifies crowds based on size and violence level. The proposed system incorporates end-to-end security by creating a Dynamic Multipoint Virtual Private Network (DMVPN) and leverages the property of IP Security (IPSec) and Firewall for confidentiality and integrity during transmission and storage. Experiment analysis and real-time inference using DeepStream Software Development Kit (SDK) proved that the proposed system has significant implications for public safety, security, and crowd management in various contexts, including public spaces, transportation hubs, and large-scale events.

INDEX TERMS Crowd behavior recognition, deep learning, public safety, secure data transmission, smart surveillance, system design.

I. INTRODUCTION

Surveillance cameras are widely used to monitor actions and detect concerning behavior and have been used by government entities, law enforcement, and private entities to monitor certain areas and geographical regions and initiate appropriate responses based on actions observed. This advantage has led to the use of large numbers of surveillance camera systems to be implemented in different countries. For instance, China, the United States of America, and the United Kingdom have deployed around 15 million, 112 thousand, and 628 thousand Closed Circuit Television (CCTV) cameras, respectively [1].

Despite the success of surveillance cameras, as evidenced by their wide use, their utilization still suffers from a major drawback. Conventional use of surveillance cameras relies on human operators who monitor footage coming from surveillance cameras and alert authorities if they detect concerning events. This means that a great number of people are required to operate large networks of surveillance cameras. If an insufficient number of operators are allocated for monitoring surveillance footage, critical events could be left undetected. Additionally, although CCTV has dramatically benefited many different areas (i.e., crime and safety monitoring,

The associate editor coordinating the review of this manuscript and approving it for publication was S. M. Abdur Razzak.

theft and vandalism detection, etc.), it is still a *reactive* approach when it comes to public safety monitoring. With more than half of the world's population residing in cities, the need for smarter insight into the city's workings is imperative. As cities become more crowded, public safety due to disasters, unrest, public gatherings, crimes, etc. becomes an ever-rising issue. Therefore, crowd detection and assessment are becoming an integral part of any city (both for safety and planning). Due to the growth of cities, traditional means of city surveillance and protection are insufficient. It becomes increasingly evident that the answer to developing smarter and safer cities lies largely in surveillance data analytics. Developing a system that can detect crowds, understand their behaviors, and develop methods to effectively manage them is still a scattered effort, despite the potential benefits.

Several systems of real-time video analysis already exist in the market. For instance, Senstar Corporation [2] developed many smart security and video management systems. One such system developed by Senstar, meant for security applications, is a crowd detection system [3] that estimates the number of people captured by CCTV cameras and sets off an alarm when a certain capacity or percentage of occupancy is reached. Another security-focused smart surveillance application that is widely used with facial recognition is to detect certain individuals. For example, Brondby, a Danish Football club, uses a facial recognition system in their stadium to identify fans who are banned from attending games due to previous unruly behaviour [4]. Amazon uses a video-analysis system named "Just Walk Out" [5], that tracks customers inside their convenience store chain, Amazon Go, to automatically identify how much to charge each customer, eliminating the need for long checkout lines.

Given the state-of-the-art, there does not exist a system designed to provide an autonomous and proactive approach to crowd surveillance and behavior/event detection. Furthermore, no system in the literature provides secure data transfer which is a necessity to preserve the integrity and authenticity of data, to prevent unauthorized access and manipulation, and to protect the privacy of individuals. As such, we propose the design of a secure, intelligent, and proactive system, called **PublicVision**, that combines the rich capabilities of Artificial Intelligence (AI) to advance the capabilities of government and municipal agencies to manage critical public safety and plan city services accordingly. The general infrastructure of the proposed **PublicVision** system is shown in Figure 1. The system comprises three layers 1) Source Spoke Layer, 2) Secure Transportation Layer, and 3) Central Hub Layer. The geographically located CCTV cameras and corresponding connected routers are the main components of the source spoke layer. The central hub layer is responsible for running the Deep Learning (DL) model on the footage coming from each CCTV camera in real-time while the Secure Transportation Layer provides the security to the data using a Virtual Private Network (VPN) and firewall.

Specifically, we design and build a system that automates city-wide surveillance, automatically detecting family of concerning events and alerting authorities about the location, nature, and extent of the behavior observed. We are specifically interested in crowd behavior detection as it is a crucial task that is especially important during periods of social unrest and large public events. Unlike action recognition tasks in the literature, we are interested in capturing information about both the size and behavior of a crowd. A training dataset that fits our purposes does not exist in the literature. Thus, we initiated a data collection effort focused on developing a dataset that encompasses various public scenery (i.e., crowds of different sizes and violence levels).

The proposed **PublicVision** primarily focuses on the automatic detection of crowd behavior, leveraging the capabilities of Deep Learning (DL) techniques. These techniques are prominent nowadays to detect human actions [6], detect and segment objects [7], classify images [8], and so on. In all cases, deep learning systems have exhibited excellent performance by automatically diving into the enviable depiction of high-level data representations. The capability of deep networks was exploited in the detection of crowd behaviors as well [9], [10], [11], [12].

In particular, our system exerted the potential of a CNN-based vision transformer, namely the Swin Transformer [13], for crowd behavior detection. Besides. we take advantage of Nvidia's DeepStream Software Development Kit (SDK) [14] which is an intelligent application framework to process real-time video data. DeepStream is a streaming analytics toolkit that can run inference on a video stream given a DL model. We use DeepStream, coupled with a DL model that we develop using the aforementioned video dataset, to run real-time inference from a central hub on footage captured by remotely placed surveillance cameras (Details are given in Section III C).

The main contributions of this work are as follows:

- An end-to-end secure smart surveillance system is devised for tracking crowd events during periods of unrest and in large public events.
- A three-layer infrastructure is built, which can ensure real-time data capturing on one end, secure communication in the middle, and smart detection of crowd behavior using AI on the other end for intelligent and real-time surveillance.
- We developed a novel video dataset and defined four distinct crowd behaviors based on factors like crowd size and violence. The automated detection of crowd behavior was achieved by training a DL model using the Swin Transformer.
- Experiments are conducted using the DeepStream SDK to ensure that our proposed system can be used in a real surveillance environment.

The remainder of the paper is organized as follows: Section II outlines the surveillance systems in the literature, previous work done in the field of video analysis, and provides details of existing human-action datasets. Section III

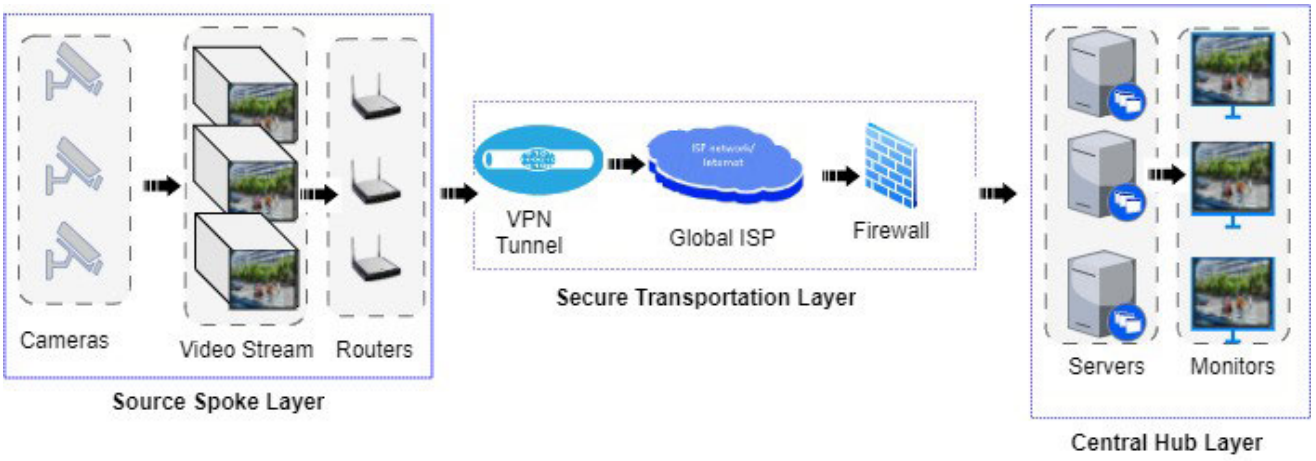

FIGURE 1. General infrastructure of the proposed **PublicVision** system.

discusses the proposed **PublicVision** system that explains the functional components and its end-to-end integration. Section IV outlines the steps taken to collect our novel video dataset and DL model development with **PublicVision** implementation details, followed by Section V that discusses the potential impact of the **PublicVision** system. Finally, the conclusion is presented in Section VI.

II. RELATED WORKS

Smart surveillance systems demand the proactive recognition and detection of events to avoid mishaps and disasters. Over the past decade, the increase in disasters and large-scale incidents during protests has prompted researchers to analyze surveillance video data using AI approaches. Besides, this led to the creation of datasets for research purposes. This section provides an outlook on the advances in video analysis and gives an awareness of existing surveillance systems and datasets.

A. SURVEILLANCE SYSTEMS

For the past two decades, with the upsurge in urban growth and urban population, CCTVs have become an essential commodity for public surveillance. In most traditional surveillance systems, captured video footage is analyzed manually, resulting in reactive rather than proactive decisions. Later, the evolution of advanced visual sensors and AI algorithms enables proactive decision-making feasible.

One of the earliest smart surveillance systems was developed by IBM for detecting activities such as suspicious behavior in parking lots, face recognition, license plate recognition, and badge identification for access control [15]. This system used the Middleware for Large Scale Surveillance (MILS) integrated with web services for data management. Another system by Fernandez et al [16]. collected data from large numbers of Internet of Things (IoT)-based visual sensors to augment the emergency team with video stream distribution and alarms. Here the data provided by visual sensors were in the form of XML, which was used to generate a semantic engine with knowledge-based ontology for vehicle route detection and abnormal trajectory tracing. Vehicle details were also analyzed in [17] to recognize vehicle make and model, color, and license plate. Low-level feature analysis from video data such as Speeded Up Robust Feature (SURF) descriptors detected the make and model while the Tesseract Optical Character Recognition (OCR) tool recognized the license plate. In [18], violations of traffic rules such as speed limit crossing, illegal parking detection, one-way violation, etc., were detected using a distributed wireless smart camera network. The distributed cameras were considered agents, and they communicated with each other via rule-based techniques to detect violations.

Besides, smart camera vendors provided surveillance solutions for traffic violations [19], [20], gunshot detection [21], loitering detection [19], [20], license plate recognition [19], [20], and suspicious human behaviors such as fighting, running, and falling [22]. The inception of smart cities also compelled smart surveillance systems deployment for intelligent traffic monitoring [19], [20], abandoned object detection [19], radioactive isotope detection [20], and intelligent routing [20]. Even though researchers and vendors provide systems for traffic monitoring and other smart city applications, none of them except the approaches of [22] and [23] can be used for the analysis of crowd behavior. However, [22] fails to address complex behaviors when crowd density increases, whereas [23] lacks experimental analysis in a real environment.

In a surveillance system, analysis and detection of crowd behavior have emerged as a prominent topic as law enforcement authorities and security personnel face many challenges due to crowd gatherings in public places. Even though in the field of computer vision, many works [9], [10], [11], [12] were there for crowd analysis, none of them provide an end-to-end solution for crowd management in real-time. Besides, none of the studies consider the security

aspects of data transmission, a significant aspect in today's world. Hence, real-time detection of crowd behavior and secure data transmission is inevitable to make reliable smart surveillance systems for critical decisions that help prevent probable crowd-related accidents and abnormal activities.

B. ADVANCES IN VIDEO ANALYSIS

Over the past several years, significant advancements have been made in video analytics using DL [24]. Specifically, several works have tackled Human Activity Recognition (HAR) [25], [26], [27], which is the task of recognizing certain human actions from a series of image frames. Attention has been drawn to HAR after several DL techniques were shown to be useful for video analysis tasks. Tran et al. [28] first proposed inflating two-dimensional Convolutional Neural Networks (2D CNNs) into three-dimensional Convolutional Neural Networks (3D CNNs). 3D CNNs are able to learn spatiotemporal features, which are capable of processing series of frames, or videos. Carreir and Zisserman [29] also proposed a Two-Stream Inflated 3D (I3D) ConvNet, which inflates the usual 2D ConvNets into 3D ConvNets for video analysis. Carreir and Zisserman test I3D on the Kinetics video dataset [30]. 3D CNNs were then shown to suffer from short-term memory; they are only capable of learning from 1 to 16 frames [31]. As a result, Shi et al. [32] proposed Convolutional Long short-term memory (Convolutional LSTMs) networks, a variant of Recurrent Neural Networks (RNNs). Convolutional LSTMs replace the fully-connected input-to-state and state-to-state transitions of conventional LSTMs, a variant of RNNs, with convolutional transitions that allow for the encoding of spatial features.

Recently, transformer-based architectures have attracted significant attention. Transformers use self-attention to learn relationships between elements in sequences, which allows for attending to long-term dependencies relative to RNNs, which process elements iteratively. Furthermore, transformers are also more scalable to very large capacity models [33]. Finally, transformers assume less prior knowledge about the structure of the problem as compared to CNNs and RNNs [34], [35], [36]. These advantages have led to their success in many computer vision tasks such as image recognition [37], [38] and object detection [39], [40]. Dosovitskiy et al. [37] proposed ViT, which achieved promising results in image classification tasks by modeling the relationship (attention) between the spatial patches of an image using the standard transformer encoder [41]. After ViT, many transformer-based video recognition methods [13], [42], [43], [44] have been proposed. In these works, different techniques have been developed for temporal attention as well as spatial attention.

In a nutshell, transformer-based approaches have led to significant advancements in the realm of computer vision. The performance improvements are quite impressive and represent a major step forward in this field. Among the transformer frameworks discussed above, the Swin Transformer [13] has really been a game changer in the field of computer vision. It has set new records in object detection [13] and semantic segmentation benchmarks [13], and has shown that transformer approaches are the future of visual modeling. In addition, Swin Transformer possesses shifted non-overlapping windows, which makes it suitable for faster running speed and hardware friendly, which inspired us to use the framework as the backbone of our proposed model (Details of Swin Transformer framework are given in Section IV-B).

C. EXISTING DATASETS

Early video datasets for action recognition include the Hollywood [45], UCF101 [46], UCF50 [47], and the HMDB-51 [48] dataset. The Hollywood dataset provides annotated movie clips. Each clip in the dataset belongs to one of 51 classes, including "push", "sit", "clap", "eat", and "walk", while the UCF50 and UCF101 datasets consist of YouTube clips grouped into one of 50 and 101 action categories, respectively. Examples of action classes in the UCF50 dataset include "Basketball Shooting" and "Pull Ups" while the action classes in UCF101 include a wider spectrum of classes subdivided into five different categories, namely, body motion, human-human interactions, human-object interactions, and playing musical instruments and sports. The Kinetics datasets [30], [49], [50], more recent benchmarks, significantly increase the number of classes from prior action classification datasets to 400, 600, and 700 action classes, respectively. The aforementioned pre-existing datasets are useful for testing different DL architectures but are not necessarily useful for specific practical tasks, such as surveillance, which likely require the distinction between a limited number of specific action classes.

In terms of public datasets that encompass violent scenery, a dataset focused on violence detection in movies is proposed by Demarty et al. [51]. Movie clips in this dataset are annotated as violent or non-violent scenes. Nievas et al. [52] introduce a database of 1000 videos divided into two groups, namely, fights and non-fights. Hassner et al. [53] propose the Violent Flows dataset, which focuses on crowd violence and contains two classes; violence and non-violence. Sultani et al. [54] collected the UCF-Crime dataset, which includes clips of fighting among other crime classes (e.g., road accident, burglary, robbery, etc.).

Perez et al. [55] proposed CCTV-fights, a dataset of 1000 videos, whose accumulative length exceeds 8 hours of real fights caught by CCTV cameras. Akti et al. [56] put forward a dataset of 300 videos divided equally into two classes; fight and non-fight. UBI-fights [57] is another dataset that distinguishes between fighting and non-fighting videos. The aforementioned datasets are summarized in Table 1 where the number of action classes and size of the dataset are outlined. In particular, the last column shows the number of videos and the cumulative duration of all video clips in each dataset (if this information is available).

TABLE 1. Existing action recognition and crowd datasets in the literature.

Dataset	Classes	Cardinality and Size
Hollywood [45]	8	475 videos / 1.4 hours
UCF101 [46]	101	13,000 videos / 27 hours
UCF50 [47]	50	6,682 videos / 14 hours
HMDB51 [48]	51	6,766 videos / 6 hours
Kinetics [30]	400	306,245 videos
Hockey dataset [52]	2	1000 videos
Violent Flows [53]	2	246 videos / 14 minutes
CCTV-fights [55]	2	1000 videos / 8 hours
SC Fight Dataset [56]	2	300 videos / 11 minutes
UBI-fights [57]	2	1000 videos / 80 hours
Our Dataset	4	1413 videos / 30 hours

In short, although the HAR datasets are useful for testing different DL architectures, they are not necessarily useful for specific practical tasks, such as surveillance, which likely requires the distinction between a limited number of specific action classes. Furthermore, to the best of our knowledge, no video dataset in the literature contains large gatherings, such as protests, as an action class. For instance, protest datasets in the literature are limited to image datasets [58] and protest metadata [59], which document protester demands, government responses, protest location, and protester identities. Thus, the novelty of our developed video dataset is that it is specifically aimed toward identifying scenarios of public unrest (violent protests, fights, etc.) or scenarios that have the potential to develop into public unrest (large gatherings, peaceful protests, etc.). Large gatherings are particularly interesting and important to be carefully monitored as they can lead to unruly events. Large gatherings that seem peaceful can evolve into a violent scenario with fighting, destruction of property, etc. In addition, the scale of violence captured can inform the scale of the response from law enforcement. Thus, for the current task, we divide violence into small-scale violence (i.e., F) and large-scale violence (i.e., LVG). To our knowledge, these aspects have been largely neglected in existing datasets, which motivates this work.

III. PROPOSED DESIGN OF PUBLICVISION SYSTEM

The design of the proposed end-to-end surveillance system is based on general infrastructure, as illustrated in Figure 1. The infrastructure is a three-layered framework - a source spoke layer, a secure transportation layer, and a central hub layer- that enables transmission, routing, and connectivity of data. These layers encompass various hardware, software, protocols, and technologies that facilitate the efficient and reliable transfer of information between devices, systems, and users. The following subsections discuss the details of the functional components in each layer.

A. SYSTEM MODEL AND DESIGN

The model and design of the functional components associated with each layer are portrayed in detail using the schematic diagram shown in Figure 2.

1) SOURCE SPOKE LAYER

The main component in this layer is a set of networks of CCTV cameras, where each network exists in separate geographical locations. As shown in Figure 2, the cameras located in multiple geographic locations collect real-time video streams, which are then forwarded to a central location via integrated service routers (ISR). The ISR is a network router that securely connects digital networks for information transmission. In particular, a sub-network is established in this layer through an ISR router in each geographical location. Since the live footage must travel through an Internet Protocol (IP) network to the central hub, we employed IP cameras in the proposed system. Hence, we build and test our **PublicVision** system using Samsung's PNM-9020VP IP camera [60] (Refer to Figure 3). It is a multi-sensor panoramic camera with a horizontal angular field of view 180°. It supports three video compression standards-H.265, H.264, and MJPEG with a maximum frame rate of 30fps.

In our proposed **PublicVision** system, we use ISR for transmitting the video streams to the central hub layer. Also, they are used for similar branch-to-branch communication by allowing each camera's footage to communicate to the central hub. The router encapsulates data into small packets for transmission through a secure network and has added features such as mobile connectivity, cloud computing, and multimedia performance. Moreover, the ISR is configured with a dynamic multipoint virtual private network (DMVPN) and IPSec for secure data transmission. The particulars of DMVPN and IPSec are presented in the following subsections.

2) SECURE TRANSPORTATION LAYER

This layer offers secure data transmission from the source spoke layer to the central hub layer. Secure data transference in surveillance is necessary to safeguard the privacy of individuals, maintain the integrity and authenticity of data, prevent unauthorized access and tampering, and ensure the reliability of the surveillance system as a whole. In the proposed system, we make use of a VPN to secure the network, as we need to transfer data over the public Internet. However, our cameras are in multiple locations, which forced us to use Internet Protocol Security (IPSec) over DMVPN instead of standard IPSec in VPN.

DMVPN [61] is a routing solution to build VPN networks with multiple nodes. DMVPN allows any two nodes to communicate with one another without having to go through a hub. It combines IPSec encryption, generic routing encapsulation (GRE) tunnels, and Next Hop Resolution Protocol (NHRP) for secure data transmission. Since we are using DMVPN, an encryption tunnel is created using GRE between the source router (ISR) and destination (Central hub layer servers). This enables us to transmit data securely even when the underlying network is the public Internet. Also, the NHRP configuration helps to find the best route to the

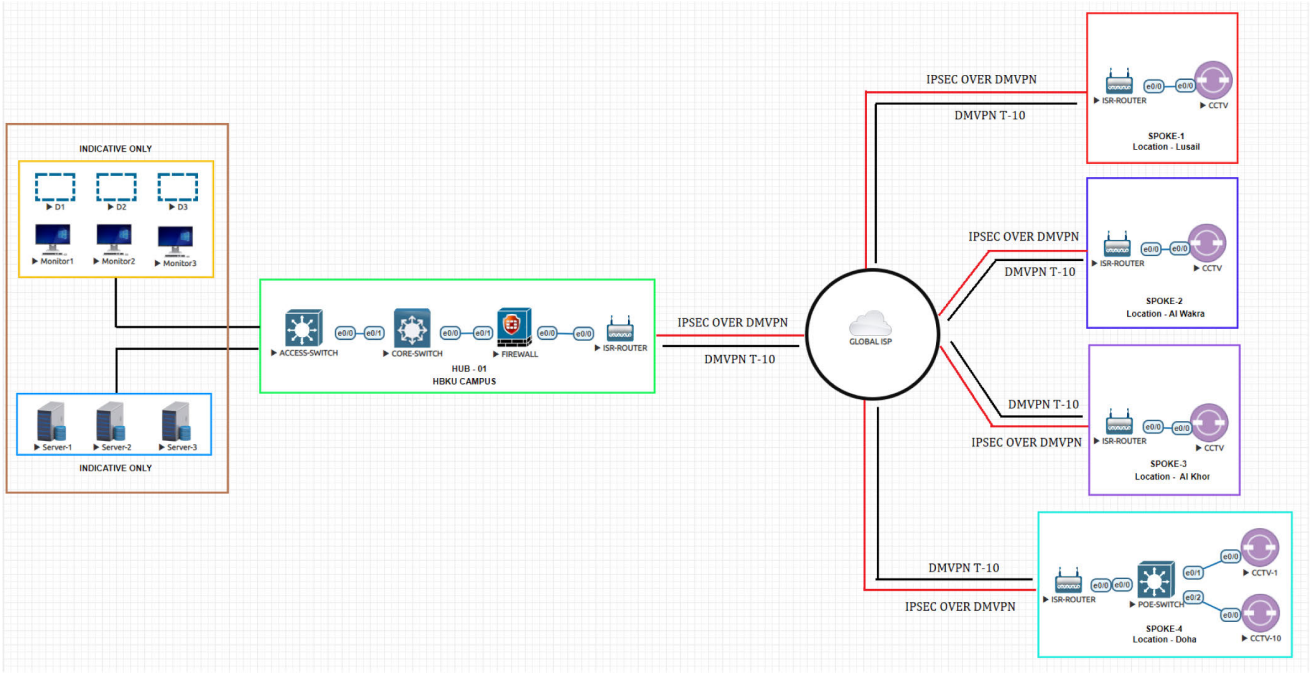

FIGURE 2. The general scheme of the proposed **PublicVision** system. The live footage of each CCTV camera at the source spoke layer is communicated in real-time to the central hub layer that runs a video-analyzing Deep Learning model on that footage. The communication between each CCTV camera network and the central hub layer is secured through DMVPN-over-IPSEC in addition to a firewall in the secure transportation layer that manages incoming traffic.

FIGURE 3. Sample picture of the camera used in the **PublicVision** system.

destination with a minimum number of hops. Here, IPSec adds an extra layer of security by providing authentication to the GRE-encrypted data packets. On the other hand, IPSec [62] is a framework that protects traffic on the network layer. The salient features include confidentiality, integrity, authentication, and replay protection. Confidentiality ensures that only the sender and receiver can read the transmitted data, while integrity guarantees that the data is not altered en route between the sender and the receiver. Authentication allows the receiver to verify that the data received originated from the claimed receiver, and replay protection protects the data from attackers capturing the video and replaying it at a later time.

Furthermore, to prevent the central hub layer from unwanted traffic, incoming traffic into the hub sub-network is controlled by a Firewall, allowing only the designated CCTV cameras to communicate with the central hub (i.e., sending their footage). The Firewall's policy is set to allow only devices with IP addresses that match the IP addresses of the CCTV cameras to send network packets into the hub's sub-network. The Access List of the Firewall is configured as the list of IP addresses of the CCTV cameras in the different geographical locations. Specifically, we use the Next Generation Firewall (NGFW), which provides complete application visibility and control, application-level awareness, threat control using sandboxing, identification services, a comprehensive set of security technologies, an integrated Intrusion Protection System (IPS), and Intrusion Detection System (IDS), and capable of decrypting and inspecting Secure Sockets Layer (SSL) for incoming and outgoing traffic.

In addition to the security measures, this layer manages the traffic between the source spoke layer and the central hub layer using two switches - the access switch and the core switch. The access switch is an Ethernet switch that connects the devices in the central hub layer with the core switch, whereas the core switch acts as a backbone transmission system between the CCTV cameras in the source spoke layer and the access switch.

3) CENTRAL HUB LAYER

The kernel of the proposed **PublicVision** system is the central hub layer which consists of GPU-equipped AI servers and display monitors. Notably, the GPU-equipped AI servers in the central hub layer are responsible for analyzing the video stream and classifying the behavior using a DL model. When a CCTV camera's footage arrives at the central hub, it is analyzed by the DL model to classify behavior in the footage into one of the four behavior classes. The behavior classes we identified were *Natural Event, Fighting, Large Peaceful Gathering, or Large Violent Gathering*. Section IV will provide the details of the DL model used and the four behavior classes to which the observed footage will be classified. To analyze incoming footage in real-time using the developed DL model, we take advantage of Deepstream [14], a software development kit developed by Nvidia.

The Deepstream SDK can be used to develop and deploy efficient visual AI applications. Deepstream allows for running a given DL model on a video stream in real time by feeding the last several frames received from the video stream to the model. The number of frames to be fed to the model at any time will depend on a pre-determined parameter, the input size of the model [14]. The output of the DL model will be one of four labels or classes of behavior. Finally, since the Deepstream requires a GPU to run [14], we employed NVIDIA GeForce RTX 2080 Ti GPUs as AI servers in the central hub layer. Besides, effective surveillance can only be achieved by visualizing CCTV footage and its associated behavior. Since DeepStream has the ability to display the incoming video stream along with the label, we utilize display monitors for live footage visualization. This could also be useful for decision-making by viewing CCTV footage.

B. SYSTEM INTEGRATION

The proposed **PublicVision** system is an end-to-end solution for the behavior recognition of crowds. Based on the layered framework discussed above, we represent this end-to-end system as a directed graph, $G = \{C, E\}$, with the vertices C as the functional components in the framework and edges E as the connection between the components as shown in Figure 4. The components (nodes in the graph) are responsible for executing tasks allocated to them, whereas the edges pass relevant data between the components.

The *CCTV* camera acts as the source of the framework responsible for monitoring the area in its field of view and captures *Raw_Video_Stream*. In contrast, the *Monitor* acts as a sink to display the behavior detected frame along with its associated label. That is, the goal of this end-to-end surveillance system is to view the frame, f_i, and detected crowd behavior, b_i while providing a *Raw_Video_Stream*, $V = \{f_0, f_1, f_2, \ldots\}$, containing events of interest. The two components, *ISR Spoke* and *ISR Hub* are routers responsible for the encapsulation/decapsulation and encryption/decryption of the stream packets resulting in *Encrypted_Stream* and *Decrypted_Stream* based on DMVPN and IPSec.

The configuration process starts when a DMVPN tunnel interface is created between the *ISR Spoke* and *ISR Hub* by enabling NHRP authentication and GRE multipoint mode. The NHRP protocol facilitates the dynamic mapping of a next-hop destination address to the physical address (MAC address) of the device responsible for forwarding packets to that destination. On the other hand, multipoint GRE enables the creation of a virtual tunnel between multiple *ISR Spoke*s in the Source Spoke layer allowing for the encapsulation and transport of network traffic over an IP network. In particular, the *ISR Hub* is connected to multiple remote *ISR Spoke*s and acts as a central point that can receive and forward traffic from any remote spokes to the appropriate destination. To complete the connection configuration, appropriate routing protocols must be enabled on *ISR Spoke* and *ISR Hub*. In our implementation, we use the Enhanced Interior Gateway Routing Protocol (EIGRP) that combines the features of distance-vector and link-state routing protocols, making it a hybrid routing protocol. It uses the Diffusing Update Algorithm (DUAL) to calculate the shortest path and determine the best routes to destination networks.

Further security is ensured for the video stream by enabling IPSec in the DMVPN tunnel, which provides confidentiality, integrity, and authentication to the transmitted *Encrypted_Stream* while traversing the *Global ISP*. During the IPsec negotiation process, the *ISR Spoke* and *ISR Hub* exchange and verify a pre-shared key (PSK) using the Internet Key Exchange version 2 (IKEv2) for authentication. If the keys match, IPsec proceeds to establish a secure connection using the Triple Data Encryption Standard (3DES) for encryption and Message Digest Algorithm 5 (MD5) for integrity and authentication. 3DES is a symmetric encryption algorithm that applies the DES algorithm three times on each data block, while MD5 (Message Digest Algorithm 5) is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value which is a unique fingerprint for a given output for verifying the integrity of data. Algorithm 1 portrays a concise representation of the configuration procedure for *ISR Spoke* and *ISR Hub*.

Furthermore, to ensure that the streams come from authenticated *CCTV* cameras, the node *Core Switch* forwards the *Decrypted_Stream* to *Firewall ASAv*. The task assigned to *Firewall ASAv* is to maintain an access list to allow traffic from trusted *CCTV*. In the proposed *PublicVision*, we employed NGFW, which offers application visibility and control, an intrusion prevention system (IPS), and advanced malware protection. Finally, the *Access Switch* passes the *Permitted_Stream* to the AI *Server* that runs the DL model to detect behavior b_i. In particular, the DeepStream SDK running on the *Server* captures the stream and feeds the last twenty frames to the DL model. Subsequently, DeepStream embeds the b_i obtained from the DL model into the incoming feed and displays it as *Detected_Behavior* on the *Monitor*.

As discussed in Section III, we have developed a novel dataset to train the DL model that detects crowds based

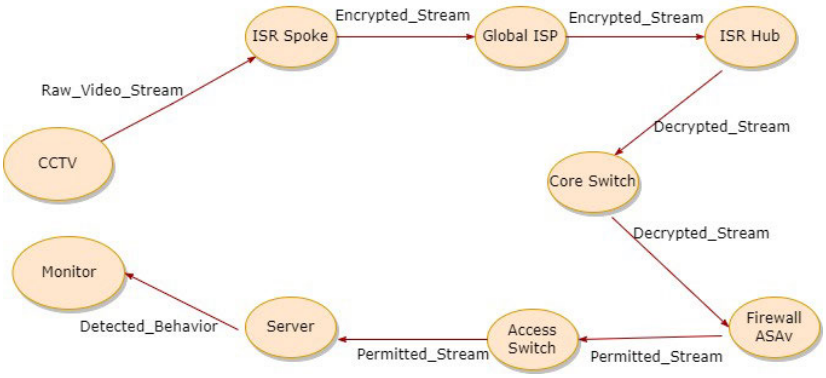

FIGURE 4. End-to-end representation of the proposed **PublicVision** system.

Algorithm 1 Configuration of *ISR Spoke* and *ISR Hub*

Notations Used: EA ← Encryption algorithm, K ← Pre-shared key
Require: *EA*, *K*
procedure ConfigISRHub(*EA*, *K*)
    Configure NHRP, GRE, and EIGRP.
    Enable IPSec and configure EA and K
end procedure
procedure ConfigISRSpoke(*EA*, *K*)
    Configure NHRP, GRE, and EIGRP.
    Register NHRP with the hub and establish mappings.
    Enable IPsec
    Configure EA and K to match the hub.
end procedure
procedure Tunnel(IP Address, EA, K)
    Initiate_Tunnel ← Encaps(IP Address)
    VPN_Tunnel ← EA(Initiate_Tunnel, K)
end procedure
procedure Spoke-Hub(DestIP)
    R_spoke ← Query (Hub_Public_IP)
    Establish direct IPsec tunnels using Hub_Public_IP.
end procedure

on size and violence level (Dataset development and other details are provided in Section IV). Thus *Server* detects b_i from *Permitted_Stream* using the trained DL model, which is redirected to *Monitor* as *Detected_Bahavior* that comprises f_i with its associated label b_i- *Natural Event (N), Fighting (F), Large Peaceful Gathering (LPG), or Large Violent Gathering (LVG).*

The DL model is a critical part of this proposed end-to-end system that starts at the CCTV camera capturing outdoor events and ends with a label describing the last two seconds of footage. Since the designed system is for city-wide or country-wide surveillance, footage from multiple cameras has to effectively and securely reach the central hub layer that runs the DL model. Additionally, since each camera in the source spoke layer has a unique ID, detected events can be easily and precisely located. As a result, the nature of the concerning b_i, as well as its location, can be detected and communicated to a central agency, such as the Ministry of Internal Affairs or Law Enforcement Forces, and an adequate response could be deployed in a timely manner.

IV. EXPERIMENT AND ANALYSIS

A. DATASET DEVELOPMENT

Since our application deals first and foremost with training a DL model to recognize certain human behaviors, a dataset must be available for training such a model. However, no satisfactory dataset exists in the literature that classifies crowd behavior based on dynamics and violence level. Recall that we seek to distinguish crowd behavior not only by the violent nature of the behavior but also by its extent. As far as we are aware, datasets in the literature distinguish only between violent and non-violent events, while we are additionally interested in the size of the crowd exhibiting the behavior. Particularly, we are interested in classifying crowd behavior along two axes, the violent nature of the crowd as well as the size of the crowd. The first class we are interested in consists of small non-violent crowds, which we classify as **Natural (N)** events. The second class of behavior that we believe is note-worthy is the class of small violent crowds, which we label as small-scale **Fighting (F)** events. Non-violent large crowds are labeled as **Large Peaceful Gathering (LPG)** events while violent large crowds are labeled as **Large Violent Gathering (LVG)** events. In order to build a DL model that can effectively distinguish between the four classes of interest (**N, F, LPG,** and **LVG**), we build a novel dataset of videos belonging to each of those four classes. Our developed dataset introduces a unique classification system, enabling the categorization of crowd behavior based on both the level of violence and the size of the crowd, distinguishing it from existing datasets. Figure 5 portrays the sample frames for each class. In particular, we gather **1,413 videos** that include one or more of the classes of interest. Most of the videos were obtained from YouTube, while other violence-detection datasets were also incorporated into our dataset. The videos go through

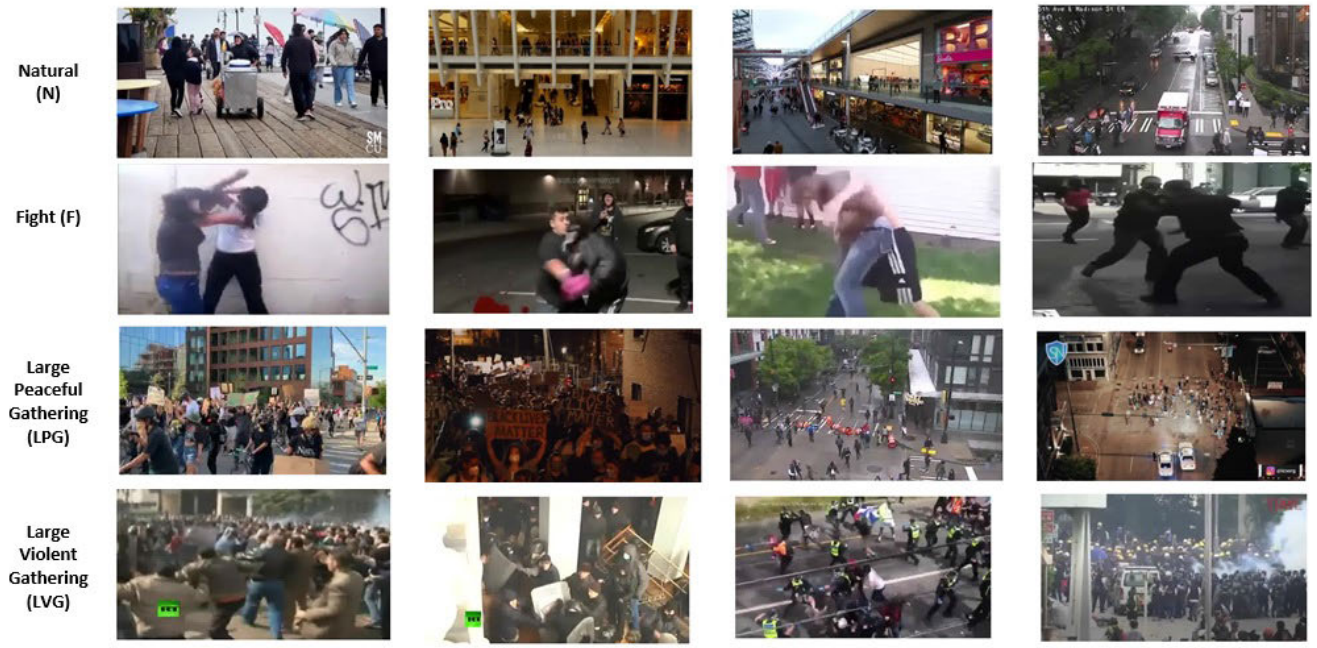

FIGURE 5. Sample frames for each behavior class from our dataset.

TABLE 2. An example annotation table describing 5 instances of the relevant classes occurring in 3 separate videos.

Video ID	Starting Time	Ending Time	Class
1	00:00:30	00:01:30	**LVG**
1	00:02:03	00:02:21	**N**
2	00:00:35	00:00:36	**LPG**
2	00:01:25	00:01:29	**F**
3	00:00:00	00:00:03	**N**

standard pre-processing steps to prepare them to be fed to a DL model for training. The subsequent subsections furnish the details of video labeling and the requisite pre-processing steps.

1) VIDEO ANNOTATION

For each video, we identify when the behaviors of interest occur. We do so by recording the start and end time stamps within which interesting behaviors are observed. The time durations wherein nothing interesting happens (no fighting, large peaceful gathering, or large violent gathering) are recorded and labeled as "baseline". The annotation process described above results in an *annotation table* such as the one shown in Table 2. Each occurrence of a class recorded in the annotation table is denoted as an *instance* of that class. Next, we will see how instances recorded in the annotation table are used to train a DL model.

2) VIDEO PRE-PROCESSING

The first pre-processing step is to unify the frame rate of all videos collected. We set the frame rate of each video to 10 frames per second (FPS). We seek to build a DL model that produces a class label based on the last 2 seconds of incoming surveillance footage. As a result, it must be trained with 2 seconds x 10 frames per second = 20-frame sequences.

After the frame rate of all videos had been set to 10 FPS, videos were broken up into their frames (10 frames for every second). Note that each video has an ordered set of frames $F = \{f_0, \ldots, f_n\}$. For each instance in the annotation table with time range $h_i : m_i : s_i - h_f : m_f : s_f$ and class b_i extracted from video V, we extract sets of 20 frames, where each set of 20 frames is called a *sample*. Samples are used to train and validate a DL model. To extract samples from an instance whose time range is $h_i : m_i : s_i - h_f : m_f : s_f$, we first identify the subset of consecutive frames $F_{instance} \in F$ that is observed during the time range of the instance $h_i : m_i : s_i - h_f : m_f : s_f$. Note that, since the frame rate of the videos was set to 10 FPS, frames $\{f_0, \ldots, f_9\}$ occur between times $0:0:0$ and $0:0:1$, frames $\{f_{10}, \ldots, f_{19}\}$ occur between times $0:0:1$ and $0:0:2$, and so on. In general, to find the first and last frames in $F_{instance}$, $f^0_{instance}$ and $f^k_{instance}$ respectively, for an instance with time range $h_i : m_i : s_i - h_f : m_f : s_f$, we apply the following formula:

$$f^0_{instance} = f_p \tag{1}$$
$$f^k_{instance} = f_q \tag{2}$$

where

$$p = 10(3600h_i + 60m_i + s_i) \tag{3}$$
$$q = 10(3600h_f + 60m_f + s_f) + 9 \tag{4}$$

Given the frames of the instance $F_{instance} = \{f_p, \ldots, f_q\}$, we can use all sets of 20 consecutive frames in $F_{instance}$, $\{f_p, \ldots, f_{p+19}\}$, $\{f_{p+20}, \ldots, f_{p+39}\}$, and so on, for testing and

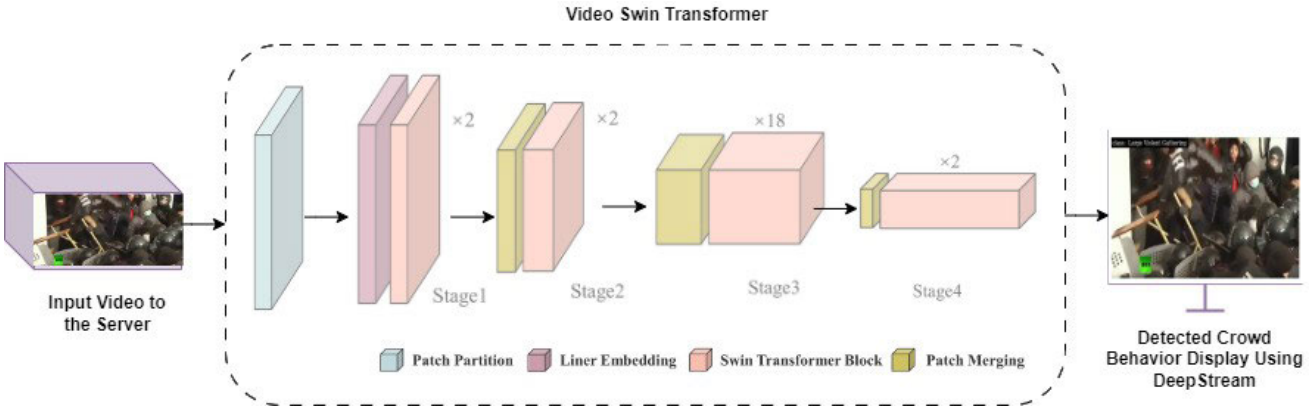

FIGURE 6. Architecture of Swin-T in the *Server* that takes input video-*Permitted_Stream* and displays the *Detected_Behavior* using DeepStream on the *Monitor*.

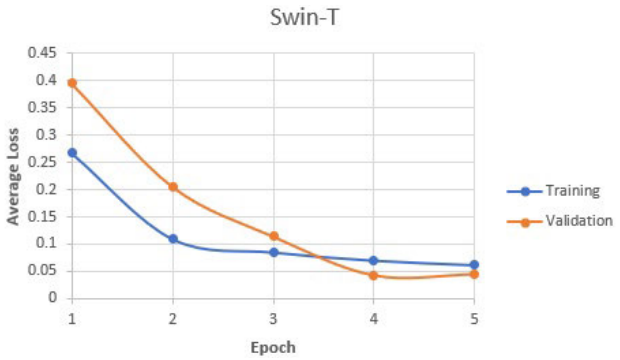

FIGURE 7. Average loss during the training and validation.

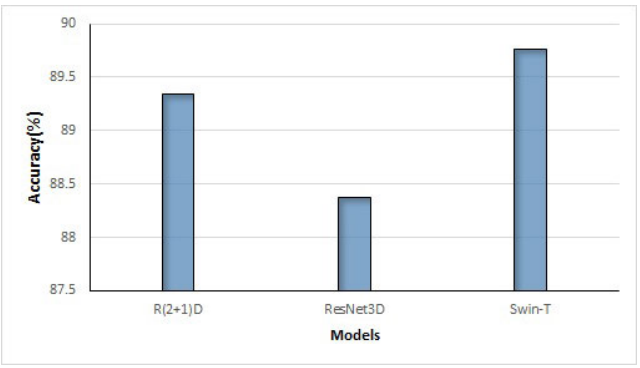

FIGURE 8. Comparison of accuracy(%) on our dataset.

validation. However, in order to avoid needlessly inflating our dataset, we skip ten frames between samples. Namely, given the frames of an instance $F_{instance} = \{f_p, \ldots, f_q\}$, we use the following sets of frames as training and validation samples: $\{f_p, \ldots, f_{p+19}\}$, $\{f_{p+30}, \ldots, f_{p+49}\}$, and so on.

B. VISION-BASED MODEL DEVELOPMENT

Having collected a dataset of samples, as illustrated in the previous section, our dataset is ready for training. In this work, we use a Swin Transformer [13], which is one of the transformer architectures that is used in many computer vision works as a general backbone for both image and video-based problems. The Swin Transformer is a hierarchical Transformer that divides images into small patches in the shallow layers of the transformer architecture and merges neighboring layers in the deeper layers to form larger patches. The Swin Transformer also utilizes shifted windows for inference, giving it greater representational power that is reflected in its recent state-of-the-art performances [13]. In addition to its state-of-the-art performance, the Swin Transformer is also more computationally effective than other models; The computation time of the Swin Transformer grows linearly with the resolution of the input images, as opposed to other models, where computation time increases quadratically with image resolution. Among multiple versions of Video Swin Transformer, we contemplate Swin-T, the tiny version of Swin as it is designed to be more efficient and faster than other versions of Swin making it well-suited for scenarios where computational resources are limited and inference speed is crucial. The overall architecture of Swin-T is provided in Figure 6.

The Swin-T framework consists of four stages, where each stage has three components- Patch Merging, Linear Embedding, and a Video Swin Transformer block except stage 1. In stage 1, each frame in the *Permitted_Stream*, $V = \{f_1, f_2, \ldots f_T\}$ is divided into 3D patches/tokens of size $2 \times 4 \times 4 \times 3$ by the 3D patch partition layer that results in $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4}$ tokens. These tokens are given to the linear embedding layer, where the features of each token are projected to an arbitrary dimension, C (For Swin-T, $C = 96$). The patch merging layers of each stage perform the spatial downsampling and concatenation of 2×2 neighboring patches, where a linear layer is utilized to project the concatenated patches to half of the input dimension. The significant block in each stage is the video swin transformer block that comprises a 2-layer multi-layer perceptron (MLP) with Gaussian Error Linear Unit (GELU) activation unit and 3D shifted window-based multi-head self-attention (3DWMSA) module.

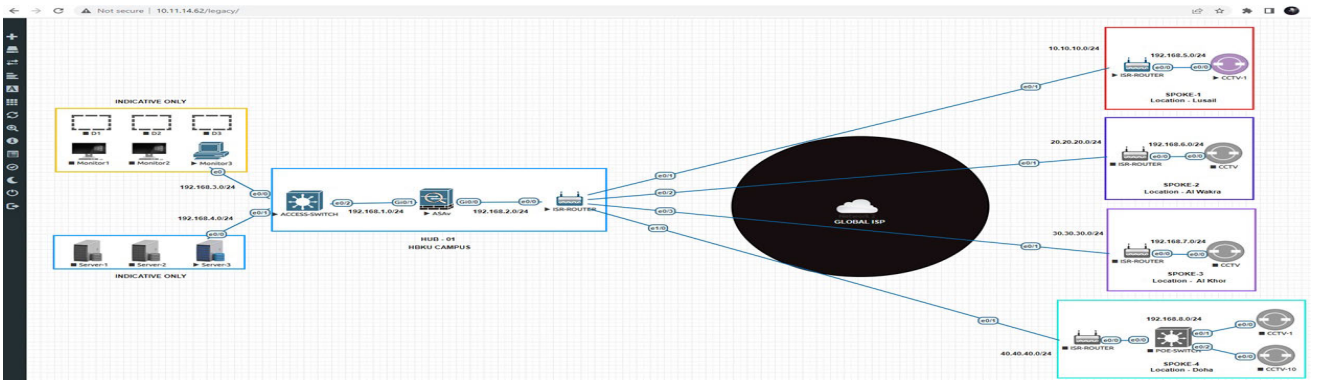

FIGURE 9. Connectivity established between SPOKE-1 and central hub layer for configuration verification.

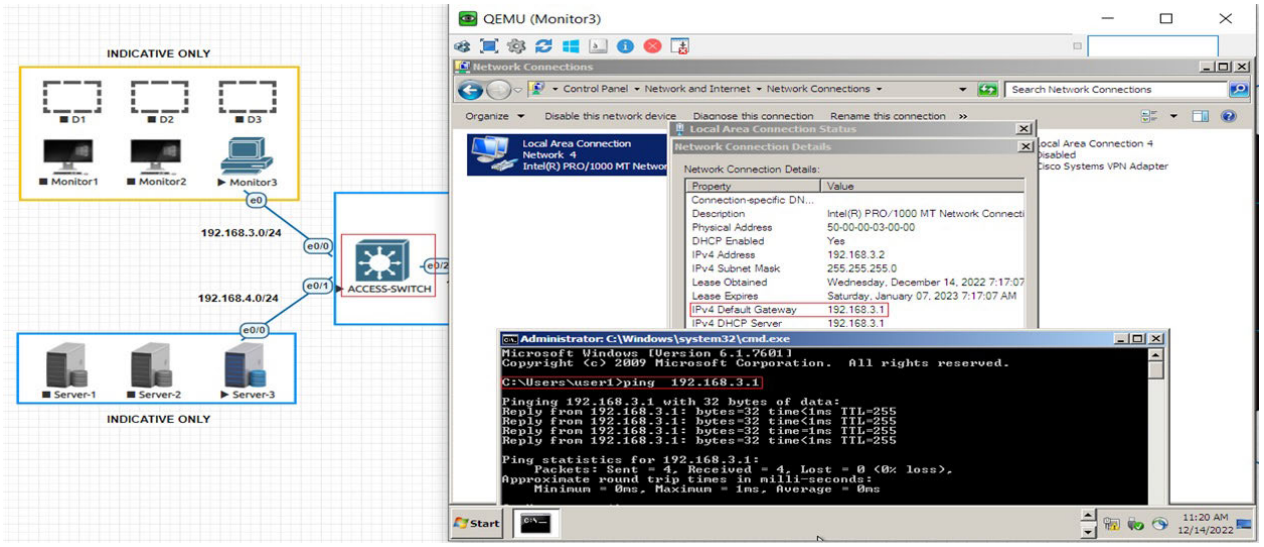

FIGURE 10. Verification of the connection between Monitor3 (IP address 192.168.3.2) and Gateway IP (IP Address 192.168.3.1).

Before passing samples through the Swin Transformer model, each frame is converted to 224 × 224 pixels, and a 20 × 3 × 224 × 224 tensor is built for each sample (20 RGB frames per sample, each frame is of size 224 × 224 pixels). To validate our dataset, we split it into training and validation sets. We seek to use 80% of samples for training and 20% for validation. However, to ensure that there's no correlation between training and validation samples, the samples from any one of the 1,413 videos are used either exclusively for training or exclusively for validation. To achieve such a split, and have it approximate an 80%-20% sample split as closely as possible, a simple random search approach is used to generate random training and validation video sets. At every iteration, the number of samples per class for the training and validation sets is counted. After a set amount of time, 60 seconds in our case, the best split is adopted and used for training and validation. Note that the best split is the one closest to 80:20 per-class split. The split used to train and validate the Swin transformer model is summarized in Table 3.

TABLE 3. Number of samples per class used for training and validation.

Class	Training Samples	Validation Samples
N	23,152	5,889
LPG	27,952	7,418
LVG	6,478	1,618
F	6,584	1,667
Total	64,166	16,592

The training process of the model was performed by minimizing the categorical cross-entropy loss by utilizing the optimizer Stochastic Gradient Descent (SGD) with an initial learning rate of 0.0001, a momentum of 0.9, and a weight decay of 0.0001. The hyperparameters used for training are compiled in Table 4. Figure 7 depicts the average loss values during the training and validation of crowd behavior classification. The decreasing behavior detection loss demonstrates that the proposed approach successfully detects the correct behaviors similar to the ground truth labels. The training was performed using Python's PyTorch

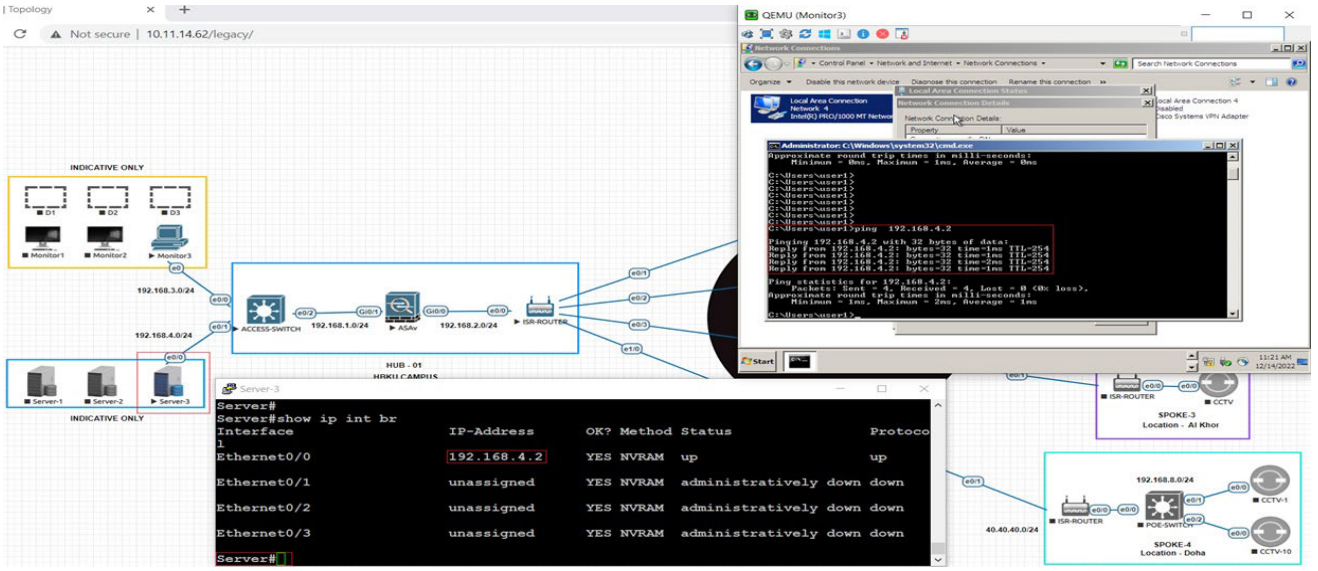

FIGURE 11. Verification of the connection between Monitor3 (IP address 192.168.3.2) and Server3(IP Address 192.168.4.2).

TABLE 4. Hyperparameters used for training Swin-T.

Hyperparameters	Values
Input Size	224×224
Initial learning rate	0.0001
Learning Rate Update Frequency	0.3
Momentum	0.9
Batch Size	16
Weight Decay	0.0001
No: of frames in a sample	20

TABLE 5. Results obtained by training the Swin transformer model on our dataset.

Class	Class Accuracy
N	93.67%
LPG	91.79%
LVG	78.74%
F	90.22%

framework in a GPU having NVIDIA GeForce with CUDA 11.4. We validated the trained Swin model by calculating the accuracy value and mean average precision(mAP) and it is observed that we attain an overall accuracy of **89.76%** and an mAP of **93.3%**. The results obtained for individual behavior classes are outlined in Table 5, where the accuracy of the model when tested using the validation samples of each class is reported. We also compare the overall accuracy of the Swin Transformer model with the ResNet3D [63] and R(2+1)D [63] frameworks, and the results (Figure 8) show that Swin Transformer has higher accuracy, which enables us to use it as the DL model for our experiments.

C. SYSTEM IMPLEMENTATION

The traffic flow and working concept of our system via DMVPN were tested using the Qemu emulator [64]. We design the network based on the general infrastructure shown in Figure 1. As per the scheme portrayed in Figure 2, CCTV cameras were installed in multiple geographical locations depicted as SPOKE-1, SPOKE-2, SPOKE-3, and SPOKE-4. The remote CCTVs in the four spokes were connected to the control room in the central hub layer via the Internet. The communication between the control room and remote CCTV was secured using DMVPN. We performed the verification of configuration and connection settings between SPOKE-1 and the central hub layer, i.e., with Server3 and Monitor3 (Figure 9). This was done to ensure that the design is correct and that the traffic of interest from CCTV passes through the VPN tunnel and has full interconnectivity between the server, storage, monitor, and remote CCTV.

The verification of network establishment through the Dynamic Host Configuration Protocol (DHCP) server configured on Access Switch was done by pinging the Gateway IP from Monitor3 as illustrated in Figure 10. Besides, we can ping our Server3 and our SPOKE-1 CCTV from Monitor-3, which ensures that there is reachability, and we can easily access the CCTV as portrayed in Figures 11 and 12.

To ensure that the DMVPN tunnel is up between the SPOKE-1 and central hub and that the traffic flow is encrypted, we run the command *show int Tunnel* in the ISR HUB and ISR SPOKE terminals. The results are displayed in Figure 13 and 14. Also, we make sure that the stream flow is defined under the dynamic routing protocol EIGRP and that neighborship is there between the VPN tunnel and IP. We also provide extra security to the system by providing a Firewall in the secure transportation layer. The hit count for incoming traffic shown in Figure 15 proves that our system guarantees that the access list is configured globally for allowing traffic from lower security level to higher security level.

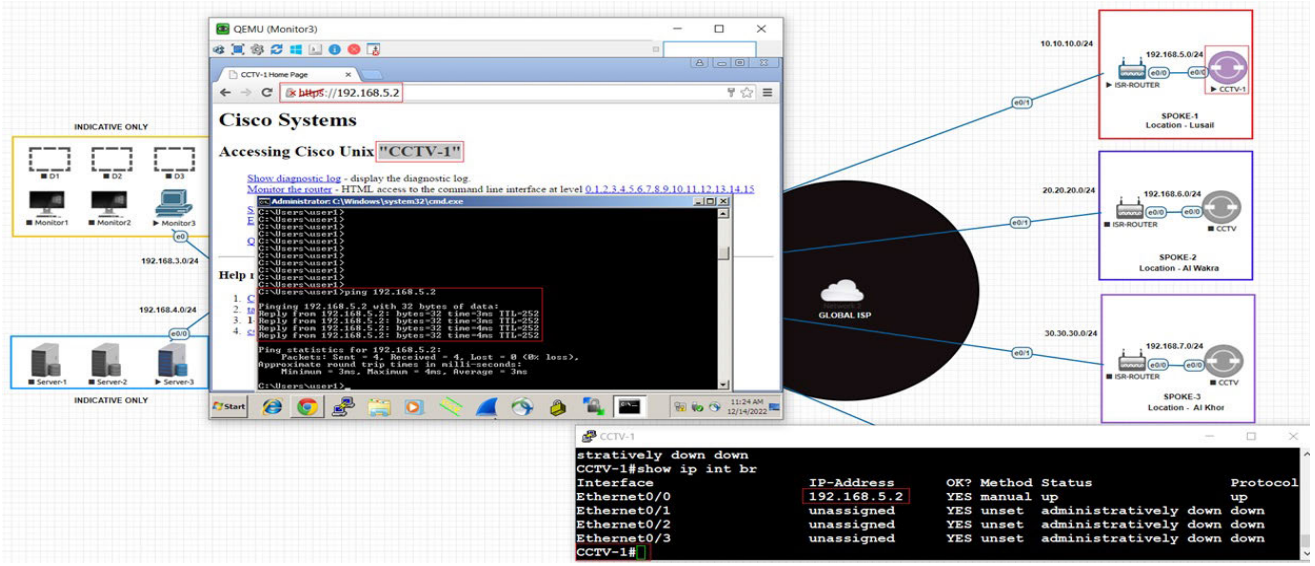

FIGURE 12. Verification of the connection between Monitor3 (IP address 192.168.3.2) and SPOKE-1 CCTV (IP Address 192.168.5.2).

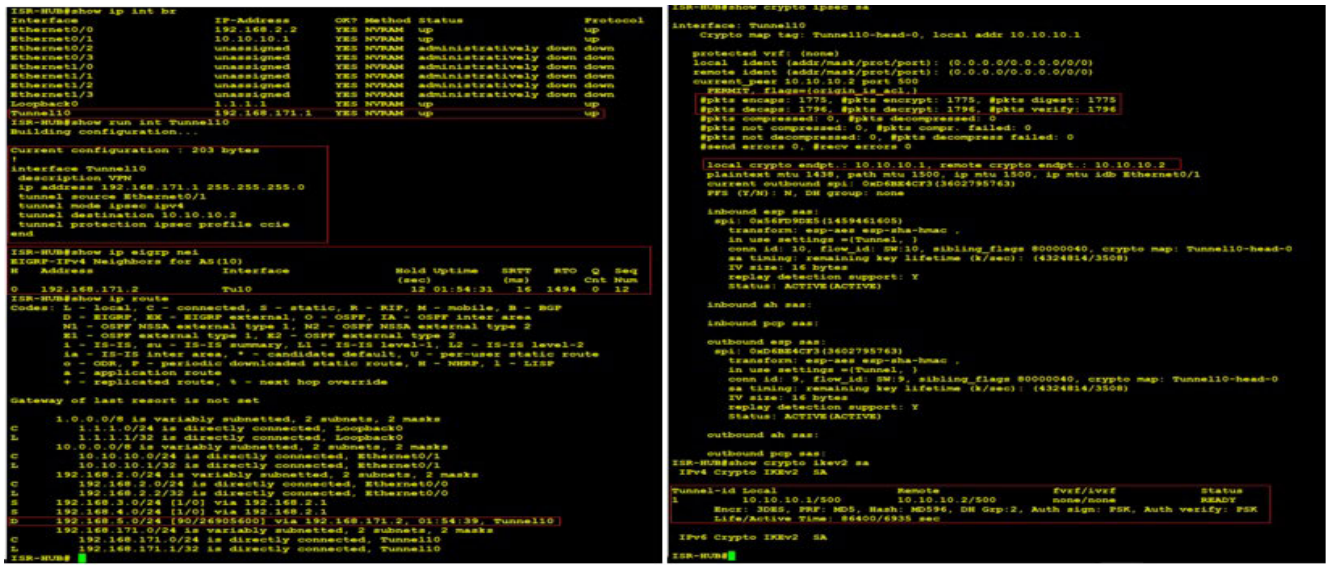

FIGURE 13. DMVPN tunnel established in ISR HUB using EIGRP, IPSEC, and 3DES encryption.

Accordingly, connectivity is established from Monitor3, located at the central hub layer, to the remote source spoke layer. We can ping and access CCTV-1 over the user interface at Monitor3, and the streaming traffic is traversing through the VPN tunnel. The traffic is encrypted/encapsulated and decrypted/decapsulated on both sides at the ISR Router. Similarly, all connections are verified and checked for the complete implementation of the system.

We design and implement an end-to-end architecture that uses IP surveillance cameras whose footage is encrypted and fed in real-time to a remote server equipped with Deepstream. Deepstream uses the Swin transformer model that we trained on a manually collected dataset. Results (Table 5) from the developed model demonstrate considerable potential for the widespread deployment of the system described in this work by security agencies. The developed DL model was converted to the ONNX format [65], which established open standards for representing machine learning algorithms and a streaming source for the video feed. We utilized pyTorch's "onnx" module to convert the model to ONNX format. The ONNX file of the DL model was specified in DeepStream's configuration file to produce an inference engine file for use in future runs of the DeepStream SDK. Figure 16 shows examples of output displayed on the monitor screen with their associated behaviors.

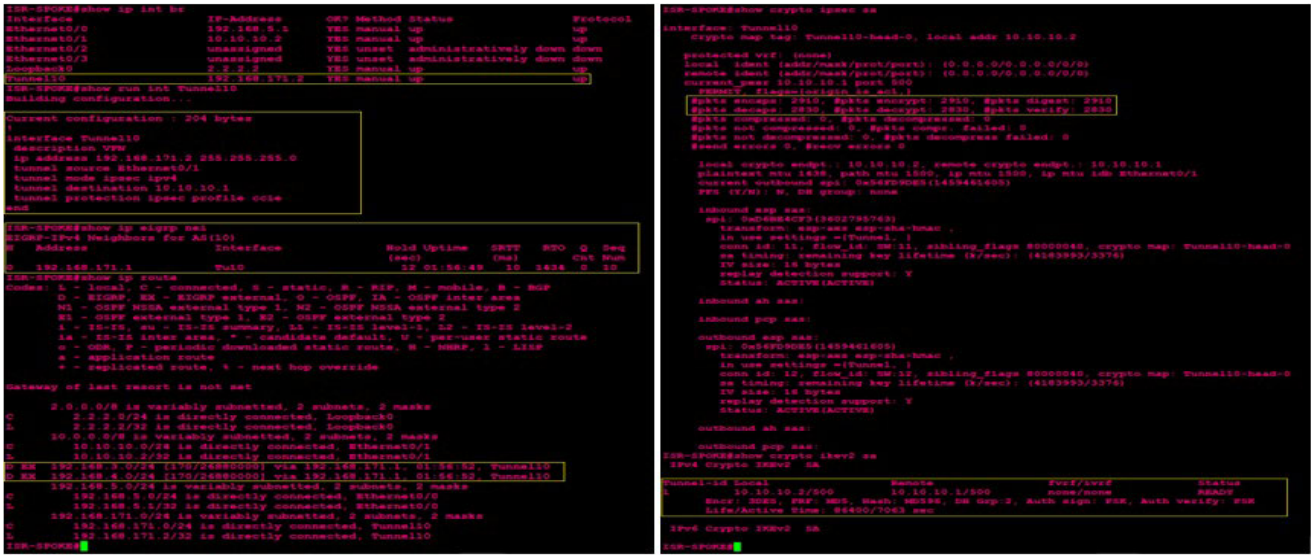

FIGURE 14. DMVPN tunnel established in ISR SPOKE using EIGRP, IPSEC, and 3DES encryption.

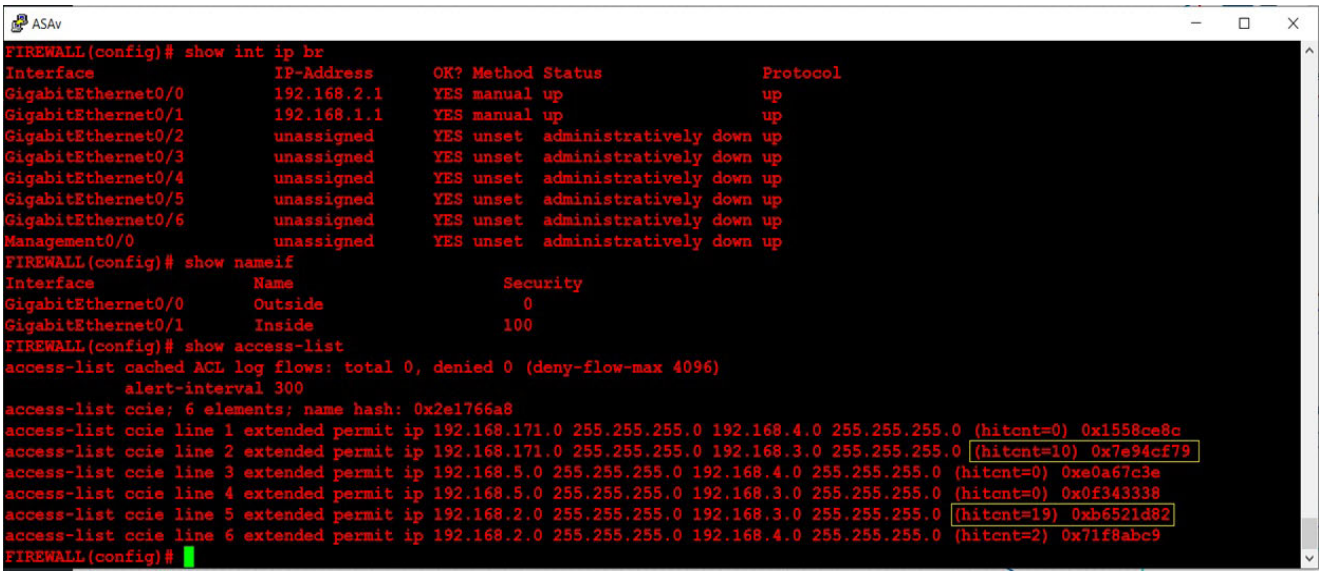

FIGURE 15. Firewall access list showing hit count.

D. COMPARISON WITH STATE-OF-THE-ART SYSTEMS

We compared the significant characteristics of our system with state-of-the-art end-to-end smart surveillance systems in the literature closely related to our work, and are displayed in Table 6. Surveillance system for crowd behavior detection based on size and violence level offers various benefits, particularly in the context of security, public safety, and event management. In particular, considering the system's characteristics, such as the usage of the DL model, the detection of crowd behavior that can distinguish the size and violence level, end-to-end secure data transmission, and real-time inference, has several advantages. These include early threat detection for proactive intervention before situations escalate, enhanced security for preventing incidents like riots, stampedes, or terrorist attacks, resource optimization, public safety, real-time monitoring to prevent tragedies, and balancing security with individual privacy rights. Table 6 clearly portrays that our system is efficient enough to handle such emergency situations related to crowds compared to state-of-the-art surveillance systems.

V. IMPACT AND APPLICATIONS

The use of the system outlined in this paper promises to have tremendous benefits for city-wide surveillance. As previously mentioned, such a system solves the major drawbacks of

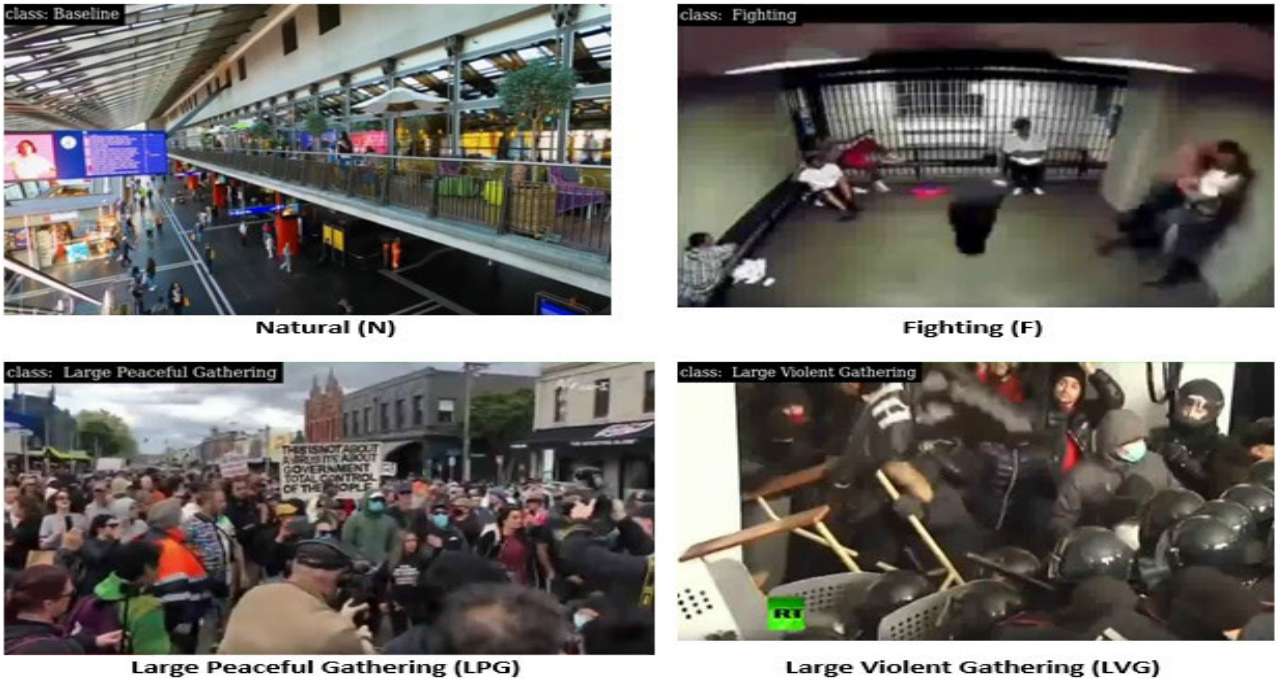

FIGURE 16. Examples of output displayed on the monitor using Deepstream. The crowd behavior corresponding to the frames is displayed in the top left corner of each frame. The class baseline depicts the Natural(N) crowd.

TABLE 6. Comparison of characteristics with state-of-the-art systems for smart surveillance.

Model	C1	C2	C3	C4	C5	C6
Tian et al. [15]	×	×	×	×	×	✓
Baran et al. [17]	✓	✓	×	×	×	✓
Eigenram et al. [18]	✓	×	×	×	×	✓
Bosch Intelligent System [19]	✓	✓	×	×	×	✓
Honeywell Smart Camera [20]	✓	✓	×	×	×	✓
Hitachi Smart Camera [21]	✓	×	×	×	×	✓
iOmniscent [22]	✓	✓	✓	×	×	✓
Varghese et al. [23]	✓	✓	✓	×	×	×
Gaikwad et al. [66]	✓	✓	×	×	×	✓
Shao et al. [67]	×	✓	×	×	×	✓
Jan et al. [68]	×	×	×	×	✓	×
Our PublicVision System	✓	✓	✓	✓	✓	✓

C1: Usage of AI models
C2: Human behavior recognition
C3: Crowd behavior detection based on dynamics
C4: Crowd behavior detection based on violence level
C5: Secure data transmission
C6: Real-time inference

human-operated surveillance systems, which are inherently expensive, in terms of the human capital required, and error-prone. The system outlined in this paper would be useful for governmental agencies all around the world, especially in cases of emergencies, such as widespread unrest, and during large-scale public events, such as concerts, national holidays, and sports tournaments. Thus, the main beneficiary of such a system would be governments all over the world. In fact, since CCTV surveillance use is already widespread in most countries, the upgrading of such systems from traditional modes of operations to the more intelligent approach described in this paper is relatively straightforward.

Governments' potential interest in the smart surveillance system outlined in this paper lies in the fact that the proposed systems allow for effective and efficient allocation of security efforts (efforts could be focused on areas where large gatherings are occurring at any point in time). This would help avoid situations getting out of hand because of a delayed or insufficient security response. Additionally, this surveillance system would allow for quick adjustment and adaptation to changing threat levels due to the fact that such a system is capable of immediately notifying authorities regarding the location, nature, and scale of note-worthy crowd behavior.

VI. CONCLUSION AND FUTURE WORK

In a public surveillance system, automated real-time analysis of crowds is often strenuous as the behavior of the crowds is unpredictable. To overcome these unforeseeable situations, datasets and models are inevitable that can recognize crowd behavior based on crowd dynamics and violence levels. Besides, surveillance systems should be reliable enough to ensure the privacy of data and should employ technical and organizational measures to safeguard sensitive information. In this context, this paper proposes **PublicVision**, an end-to-end secure surveillance system for city-wide or country-wide surveillance for crowd behavior classification based on crowd size and violence level. The proposed system consists of sub-networks of CCTV cameras whose footage is securely sent to a remote central hub, where servers will analyze incoming camera footage in real time. The DL model in the server used to analyze the camera footage is a Swin transformer model that's trained on a novel video dataset that groups crowd behavior into four categories and can distinguish crowd dynamics and violence levels. We ensure the security of the transmitted data by leveraging the implementation of DMVPN over IPSec. Experiment analysis using DeepStream SDK proves that our system is capable of real-time secure surveillance and crowd management. In the future, we are planning to create robust DL models that can fully leverage the spatiotemporal properties of video data. Additionally, we are planning to explore the integration of wireless communication protocols into the system to generate location-specific alert messages. These efforts will help enhance the capabilities of our surveillance system and improve its effectiveness in managing public safety.

ACKNOWLEDGMENT

This publication was made possible by AICC03-0324-200005 from the Qatar National Research Fund (a member of the Qatar Foundation). The findings herein reflect the work and are solely the responsibility, of the authors. Open Access funding is provided by the Qatar National Library (QNL).

REFERENCES

[1] I Global. (Jan. 26, 2022). *Role of CCTV Cameras: Public, Privacy, and Protection*. [Online]. Available: https://www.ifsecglobal.com/video-
[2] (Dec. 21, 2022). *Perimeter Security. Video Analytics. Sensors. Property Protection. (N.D.)* [Online]. Available: https://www.senstar.com/
[3] (Dec. 21, 2022). *Crowd Detection—Occupancy Detection*. [Online]. Available: https://senstar.com/products/video-analytics/crowd-detection/
[4] S. Overgaard. (Dec. 21, 2019). *A Soccer Team in Denmark is Using Facial Recognition To Stop Unruly Fans*. [Online]. Available: https://www.npr.org/2019/10/21/770280447/
[5] (Dec. 21, 2022). *Just Walk Out. (N.D.)* [Online]. Available: https://justwalkout.com/
[6] A. Hussain, T. Hussain, W. Ullah, and S. W. Baik, "Vision transformer and deep sequence learning for human activity recognition in surveillance videos," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Apr. 2022.
[7] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.
[8] Y. Li, "Research and application of deep learning in image recognition," in *Proc. IEEE 2nd Int. Conf. Power, Electron. Comput. Appl. (ICPECA)*, Jan. 2022, pp. 994–999.
[9] E. B. Varghese, S. M. Thampi, and S. Berretti, "A psychologically inspired fuzzy cognitive deep learning framework to predict crowd behavior," *IEEE Trans. Affect. Comput.*, vol. 13, no. 2, pp. 1005–1022, Apr. 2022.
[10] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: A review through deep learning techniques for crowd analysis," *J. Big Data*, vol. 6, no. 1, pp. 1–27, Dec. 2019.
[11] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection," *Inf. Fusion*, vol. 65, pp. 13–20, Jan. 2021.
[12] G. Tripathi, K. Singh, and D. K. Vishwakarma, "Convolutional neural networks for crowd behaviour analysis: A survey," *Vis. Comput.*, vol. 35, no. 5, pp. 753–776, May 2019.
[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
[14] (Nov. 21, 2022). *Deepstream Sdk. Nvidia Developer*. [Online]. Available: https://developer.nvidia.com/
[15] Y.-L. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C.-F. Shu, "IBM smart surveillance system (S3): Event based video surveillance system with an open and extensible framework," *Mach. Vis. Appl.*, vol. 19, nos. 5–6, pp. 315–327, Oct. 2008.
[16] J. Fernández, L. Calavia, C. Baladrón, J. Aguiar, B. Carro, A. Sánchez-Esguevillas, J. Alonso-López, and Z. Smilansky, "An intelligent surveillance platform for large metropolitan areas with dense sensor deployment," *Sensors*, vol. 13, no. 6, pp. 7414–7442, Jun. 2013.
[17] R. Baran, T. Rusc, and P. Fornalski, "A smart camera for the surveillance of vehicles in intelligent transportation systems," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10471–10493, Sep. 2016.
[18] D. Eigenraam and L. J. M. Rothkrantz, "A smart surveillance system of distributed smart multi cameras modelled as agents," in *Proc. Smart Cities Symp. Prague (SCSP)*, May 2016, pp. 1–6.
[19] (May 31, 2023). *Bosch Intelligent Video Analysis*. [Online]. Available: https://www.boschsecurity.com/xc/en/
[20] (May 31, 2023). *Bhubaneswar's Smart Safety City Surveillance Project Powered By Honeywell Technologies*. [Online]. Available: https://buildings.honeywell.com/content/dam/hbtbt/en/documents/downloads/Bhubaneswar-CS_0420_V2.pdf
[21] (May 31, 2023). *Hitachi: Data Integration Helps Smart Cities Fight Crime, Iot-hitachi-smart Communities-solution*. [Online]. Available: https://www.intel.com/content/dam/www/public/emea/xe/en/documents/
[22] (May 31, 2023). *Iomniscient*. [Online]. Available: https://iomni.ai/our-solutions/
[23] E. B. Varghese and S. M. Thampi, "A cognitive IoT smart surveillance framework for crowd behavior analysis," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2021, pp. 360–362.
[24] V. Sharma, M. Gupta, A. Kumar, and D. Mishra, "Video processing using deep learning techniques: A systematic literature review," *IEEE Access*, vol. 9, pp. 139489–139507, 2021.
[25] C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," *Eng. Appl. Artif. Intell.*, vol. 77, pp. 21–45, Jan. 2019.
[26] S. Arif, J. Wang, T. Ul Hassan, and Z. Fei, "3D-CNN-based fused feature maps with LSTM applied to action recognition," *Future Internet*, vol. 11, no. 2, p. 42, Feb. 2019.
[27] C.-D. Huang, C.-Y. Wang, and J.-C. Wang, "Human action recognition system for elderly and children care using three stream ConvNet," in *Proc. Int. Conf. Orange Technol. (ICOT)*, Dec. 2015, pp. 5–9.
[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[29] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Arp. 2017, pp. 6299–6308.
[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.
[31] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.
[32] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 802–810.

[33] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "GShard: Scaling giant models with conditional computation and automatic sharding," 2020, *arXiv:2006.16668*.
[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[38] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*.
[39] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[42] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.
[43] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 3156–3165.
[44] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. ICML*, 2021, vol. 2, no. 3, p. 4.
[45] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[46] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[47] *UC for Research in Computer Vision*. Accessed: May 11, 2023. [Online]. Available: https://www.crcv.ucf.edu/data/UCF50.php
[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
[49] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," 2018, *arXiv:1808.01340*.
[50] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," 2019, *arXiv:1907.06987*.
[51] C.-H. Demarty, B. Ionescu, Y.-G. Jiang, V. L. Quang, M. Schedl, and C. Penet, "Benchmarking violent scenes detection in movies," in *Proc. 12th Int. Workshop Content-Based Multimedia Indexing (CBMI)*, Jun. 2014, pp. 1–6.
[52] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Proc. 14th Int. Conf. (CAIP)*, Seville, Spain. Cham, Switzerland: Springer, Aug. 2011, pp. 332–339.
[53] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–6.
[54] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6479–6488.
[55] M. Perez, A. C. Kot, and A. Rocha, "Detection of real-world fights in surveillance videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2662–2666.
[56] S. Akti, G. A. Tataroglu, and H. K. Ekenel, "Vision-based fight detection from surveillance cameras," in *Proc. 9th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2019, pp. 1–6.
[57] B. Degardin and H. Proença, "Human activity analysis: Iterative Weak/Self-supervised learning frameworks for detecting abnormal events," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Sep. 2020, pp. 1–7.
[58] (Dec. 21, 2022). *Political Protest Movements: Data*. [Online]. Available: https://guides.library.yale.edu /c.php?g=956915amp;p=6961578
[59] (Dec. 21, 2022). *Political Protest Movements: Metadata*. [Online]. Available: https://dataverse.harvard.edu/dataverse/MMdata
[60] (Dec. 21, 2022). *Pnm-9020v*. [Online]. Available: https://www.hanwhasecurity.com/product/
[61] A. K. Farota and M. Dioum, "DMVPN (dynamic multipoint VPN): A solution for interconnection of sites IPV6 over an IPV4 transport network," *J. Telecommun.*, vol. 34, no. 1, p. 7, 2016.
[62] B. Stackpole and P. Hanrion, *Software Deployment, Updating, and Patching*. Boca Raton, FL, USA: CRC Press, 2007.
[63] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
[64] *Qemu Emulator*. Accessed: May 18, 2023. [Online]. Available: https://www.qemu.org/download
[65] J. Bai, F. Lu, and K. Zhang. (2019). *ONNX: Open Neural Network Exchange*. [Online]. Available: https://github.com/onnx/onnx
[66] B. Gaikwad and A. Karmakar, "Smart surveillance system for real-time multi-person multi-camera tracking at the edge," *J. Real-Time Image Process.*, vol. 18, no. 6, pp. 1993–2007, Dec. 2021.
[67] Z. Shao, J. Cai, and Z. Wang, "Smart monitoring cameras driven intelligent processing to big surveillance video data," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 105–116, Mar. 2018.
[68] M. A. Jan, W. Zhang, M. Usman, Z. Tan, F. Khan, and E. Luo, "SmartEdge: An end-to-end encryption framework for an edge-enabled smart city application," *J. Netw. Comput. Appl.*, vol. 137, pp. 1–10, Jul. 2019.

MARWA QARAQE (Senior Member, IEEE) received the bachelor's degree in electrical engineering from Texas A&M University, Qatar, in 2010, and the M.Sc. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, TX, USA, in August 2012 and May 2016, respectively. She is currently an Associate Professor with the Division of Information and Communication Technology, College of Science and Engineering, Hamad bin Khalifa University. Her research interests include wireless communication, signal processing, and machine learning and their application in multidisciplinary fields, including but not limited to security, the IoT, health, physical layer security, federated learning over wireless networks, machine learning for wireless communication, security, and health.

ALMIQDAD ELZEIN received the B.Sc. degree in computer engineering from Hamad bin Khalifa University, Qatar, in 2019. He is currently pursuing the Master of Applied Science degree with the Electrical and Computer Engineering Program, University of Windsor. He was a Research Associate with Carnegie Mellon University, from 2019 to 2021, and as a Research Assistant with Hamad bin Khalifa University, from August 2021 to August 2023. His research interests include optimization, deep learning, and operations research.

EMRAH BASARAN received the bachelor's degree in computer engineering from Erciyes University, Turkey, in 2011, and the M.Sc. and Ph.D. degrees in computer engineering from Istanbul Technical University, Turkey, in 2013 and 2020, respectively. He is currently a Senior Researcher with the Informatics and Information Security Research Center (BİLGEM), The Scientific and Technological Research Council of Turkey (TÜBİTAK). His research interests include image processing, computer vision, machine learning, deep learning, and TinyML.

YIN YANG (Member, IEEE) received the B.Eng. degree in computer science from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2004, and the Ph.D. degree in computer science from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, in 2009. He is currently an Associate Professor with the College of Science and Engineering, Hamad bin Khalifa University, Doha, Qatar. He has published extensively in top venues on differentially private data publication and analysis and on query authentication in outsourced databases. He is also working actively on cloud-based big-data analytics, with a focus on fast-streaming data. His main research interests include cloud computing, database security and privacy, and query optimization.

ELIZABETH B. VARGHESE received the Ph.D. degree in computer science from the Indian Institute of Information Technology and Management, Kerala (IIITM-K), Cochin University of Science and Technology, India, in 2022. She is currently a Postdoctoral Researcher with the Division of Information and Computing Technology, College of Science and Engineering, Hamad bin Khalifa University. She was awarded the prestigious Women Scientist Scheme A (WOS-A) Ph.D. Fellowship by the Department of Science and Technology (DST), Government of India. Her research interests include computer vision, deep learning, machine learning, image processing, and human–computer interaction.

WISAM COSTANDI received the B.S. degree in biomedical engineering from Northwestern University and the M.S. and M.B.A. degrees from the University of California at Davis. He is currently a Serial Entrepreneur, the Co-Founder and CEO for EMMA Systems, and the Managing Director of Informatica Qatar. He sits on the boards of various technology organizations and is the current Chair of YPO Qatar.

JACK RIZK received the bachelor's degree in electronics and communications engineering from Alexandria University, Egypt, in 2008. He is currently the Project Manager of Informatica Qatar, Division of Solutions. He is also PMP certified from PMI, Computer Science in AI from Harvard University Generative AI Overview for Project Managers from PMI. His research interests include project management, data analytics, wireless networks, wireless communications, and cyber security.

NASIM ALAM received the bachelor's degree in computer application from Jamia Hamdard University, India, in 2020. He is currently a Network and Security Engineer with Informatica Qatar, Division of Solution. He is also a Cisco Certified Network Professional (CCNP) and a Cisco Certified Network Expert (CCIE) in Security. His research interests include network design, network implementation and operation, and securing the networks.