Hindawi
Security and Communication Networks
Volume 2022, Article ID 8383461, 12 pages
https://doi.org/10.1155/2022/8383461

Research Article

Internet-of-Things-Based Suspicious Activity Recognition Using Multimodalities of Computer Vision for Smart City Security

Amjad Rehman,[1] **Tanzila Saba**,[1] **Muhammad Zeeshan Khan,**[2] **Robertas Damaševičius**,[3] **and Saeed Ali Bahaj**[4]

[1]*Artificial Intelligence & Data Analytics Lab (AIDA) CCIS Prince Sultan University, Riyadh 11586, Saudi Arabia*
[2]*Intelligent Criminology Research Lab National Center of Artificial Intelligence, KICS, University of Engineering & Technology, Lahore, Pakistan*
[3]*Faculty of Applied Mathematics, Silesian University of Technology, Gliwice 44-100, Poland*
[4]*MIS Department College of Business Administration, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia*

Correspondence should be addressed to Robertas Damaševičius; robertas.damasevicius@polsl.pl

Received 18 April 2022; Revised 17 August 2022; Accepted 29 August 2022; Published 5 October 2022

Academic Editor: Andrea Michienzi

Automatic human activity recognition is one of the milestones of smart city surveillance projects. Human activity detection and recognition aim to identify the activities based on the observations that are being performed by the subject. Hence, vision-based human activity recognition systems have a wide scope in video surveillance, health care systems, and human-computer interaction. Currently, the world is moving towards a smart and safe city concept. Automatic human activity recognition is the major challenge of smart city surveillance. The proposed research work employed fine-tuned YOLO-v4 for activity detection, whereas for classification purposes, 3D-CNN has been implemented. Besides the classification, the presented research model also leverages human-object interaction with the help of intersection over union (IOU). An Internet of Things (IoT) based architecture is implemented to take efficient and real-time decisions. The dataset of exploit classes has been taken from the UCF-Crime dataset for activity recognition. At the same time, the dataset extracted from MS-COCO for suspicious object detection is involved in human-object interaction. This research is also applied to human activity detection and recognition in the university premises for real-time suspicious activity detection and automatic alerts. The experiments have exhibited that the proposed multimodal approach achieves remarkable activity detection and recognition accuracy.

1. Introduction

In recent years, ever-increasing technological advances have made automated human activity recognition a common research subject. Video surveillance has a wide range of applications. These applications include normal and suspicious activities such as gaming, human-computer interaction, exam invigilation, detecting chaos, analyzing sports, predicting crowd behavior, etc. It is an important safety aspect for indoor and outdoor environments [1].

Innovations are occurring rapidly, and since there is a large amount of video data to process, manual intervention is not feasible and is error-prone. Additionally, it is exceedingly challenging to monitor public spaces constantly. Hence, it is necessary to install intelligent video surveillance that can track people's movements in real time, classify them as routine or exceptional, and provide alerts [2].

Human activity detection relies on sensors like radar, cameras, and cell phones to identify abnormalities in human behaviour. They are being used for human-computer interaction, surveillance, monitoring suspicious activities, and other security purposes [3, 4]. The majority of today's systems rely on video gathered from CCTV cameras. If a crime or act of violence occurs, this footage will be utilized in the investigation. It would be preferable, however, to build a system that might identify an anomalous or unexpected circumstance beforehand and notify the authorities [5, 6].

In recent years, ever-increasing technological advances have made automated human activity recognition a common research subject. Video surveillance has a wide range of applications. These applications may include normal and suspicious activities such as gaming, human-computer interaction, exam invigilation, detecting chaos, analyzing sports, predicting crowd behavior, etc. It is an important safety aspect for indoor and outdoor environments [7, 8].

Currently, innovations are occurring at a rapid pace. The most popular exploration topic these days is robotized human activity recognition. Since there is a large amount of video data to process, manual intervention will not only be tiring but also cause omissions, making the system effective and error-prone. Automatic video surveillance has tacked on this issue. It is impossible to monitor CCTV events manually. Whether the event has already occurred or not, searching for the desired event through recordings is extremely time-consuming. However, a system that automatically senses any irregular or abnormal condition in advance and alerts the appropriate authorities is more appealing. It can be used in indoor and outdoor settings [9, 10].

Different efficacious algorithms are used for automatic activity recognition on roads, airports, educational institutions, offices, etc. Computer vision has provided machines with humanlike vision. Large datasets are accessible and can be trained with GPUs' to help make future predictions. Computer vision technology has a few stages, like taking input from surveillance cameras, separating the frames, classifying and labeling the activity, and writing its description. Normally, two types of classification techniques are used in computer vision. Supervised and unsupervised; supervised classification requires manual labeling whereas unsupervised is completely computer-based and does not need computer intervention [11, 12].

Deep learning is the most exemplary architecture that learns difficult tasks among other architectures. It extracts features from images automatically and portrays significant information about the image. Since it extracts features automatically, it makes it more convenient to use. CNN learns visual patterns directly from pixels [13, 14]. Long short-term memory (LSTM) models can be used for videos as they can recall things for a longer time. The proposed work implemented the YOLOV4 for detecting the different activities related to surveillance and for recognizing the activities, 3D CNN is used. Multiple cameras are connected to the centralized system via IoT (Internet of Things) protocols. Ethernet communication creates a local server to access each camera feed through its specific IP address used in the centralized GPU for prediction [15, 16].

The remaining paper is organized as such: Section 2 explores the relevant state of art critically and highlights the need for this research. Additionally, the main contributions are also mentioned. Section 3 presents the proposed methodology; Section 4 exhibits results and analysis at length and datasets used for experiments. Finally, Section 5 concludes the research.

2. Background

Understanding human behaviour is now one of the most significant areas of computer vision research. Human activity identification uses data from sensors, such as a sequence of RGB camera images, range sensors, or other sensing modalities [17, 18], to automatically identify and understand human actions. Its applications include surveillance, video processing, robotics, and a variety of systems involving human-computer interaction [19, 20]. In the early 1980s, depth sensors improved human activity recognition. Previous research has concentrated mainly on understanding and identifying behaviors from visible light video streams. Several survey articles summarized these works at various depths and perspectives [21].

Zhu et al. [22] used motion information and contextual features for activity detection in a scene, arguing that actions have a close relationship with context. Following the identification and segmentation of behavior, a two-layered conditional random field is used to recognize events from segmented patterns and contextual knowledge. However, they did not use a benchmark dataset, nor accuracy compared to the reported literature. Yue et al. [23] compared two CNN architectures for integrating color and optical flow data for action recognition using the LSTM network. They claimed higher performance on the Sports 1 million dataset (73.1% vs. 60.9%) and the UCF-101 datasets with (88.6% vs. 88.0%) and without (88.6% vs. 88.0%) and without additional optical flow information (82.6% vs. 72.8%). Ibrahim et al. [24] came up with a two-stages of time model to look at unusual activities in the community. They made an LSTM model to show how a person acts in a series of frames, while a second LSTM network adds up the representations at the individual level. Finally, they reported an 81.5% detection accuracy. Khaire et al [25] used the CNN classifier with skeletal data to recognize the different human activities. They used two datasets and achieved 95.11% and 96.67% accuracies. Mariem et al. [26] designed a model named the history of binary motion image (HBMI). In this model, they introduced a new method for foreground detection using the Gaussian mixture model (GMM), including the Magnitude of Optical Flow (MOF). To avoid irrelevant motion, they utilized the fast frame skipping method. Hence, HBMI is a novel method of portraying instructive notions for human activity recognition based on the superposition of human shape. HBMI achieved 97.60% accuracy in the testing state. Xing et al. [27] designed a system to detect driver activities using a deep learning approach. With the help of a low-cost camera, the actions of ten drivers have been recorded. The extracted images are then segmented with a Gaussian mixture model (GMM). These preprocessed images are applied to train AlexNet [28], GoogLeNet [29], and ResNet-50 [30] on activities like texting, mirror checking, using mobile phones, etc. Among these models, AlexNet outperformed the other models with 81.6%, while GoogleNet and ResNet scored 78.6% and 74.9%, respectively. By working on static images only, Chang et al. [31] enhanced the approach of integrated 3D data of human body movements to create a three-dimensional motion history image. Xiaofei et al. [32] suggested a spatiotemporal silhouette representation to describe motion properties, including regular activities. Finally, multiclass SVM was utilized, with each operation consisting of many views and scenarios of motion descriptors. On the KTH dataset, they attained an average accuracy of 94.10%. In order to simulate the temporal distribution of players in a sporting event and foretell the future course of action, Zhong et al. [33] used hierarchical LSTMs for the temporal encoding of extracted features from video frames and trajectory data. However, no accuracy was reported.

Recently, Saba et al. [34] detected anomalies in smart hospitals by using principal component analysis (PCA) for activity feature extraction. Finally, an ensemble classifier is employed for anomaly classification. Experiments were performed on the KDDCup-“99” dataset and 93.2% accuracy was reported. Patalas-maliszewska et al. [35] adopted CNN, Support Vector Machine (SVM), and CNN region-based CNN (Yolov3 Tiny) for recognizing completed work tasks in the industrial environment. The work of González et al. [36] was heavily focused on achieving real-time results. They conducted extensive research utilizing various datasets and trained Faster-RCNN using Feature Pyramid Network with Resnet50, and outperformed by 3.91 percent as compared to reported techniques in the literature. Bhatti et al. [17] extracted data from YouTube CCTV videos/GitHub repositories and used two approaches (sliding window/classification and region proposal/object detection). They tested several pretrained deep learning classifiers; however, Yolov4 had the best performance for detecting suspicious activity, with an F1 score of 91% and an average accuracy of 91.73%.

Based on the literature analyzed, it is concluded most of the research conducted did not consider a number of expected instances, mostly used static images, and was tailored for specific purposes. However, the final aim of the proposed research is to use the automatically identified behaviors and activities in groups in live videos. Hence, in the proposed research work, we first detect the area of interest and then pass it to the classification network. The reduction of unnecessary learning information increased efficiency and accuracy.

This study has the following main contributions:

(1) A novel activity recognition and detection framework utilizing the YOLOv4 version and the 3D-CNN.

(2) Fine-tune convolution neural network architectures for better object recognition accuracy by incorporating object spatial and temporal information.

(3) Internet of Things-based architecture has been utilized to incorporate and manage the decision-making of deep learning-based architectures efficiently.

3. Proposed Methodology

The proposed methodology is based on two steps. First, we detected the region of interest (ROI) using the fine-tuned version of the Yolo-v4. Secondly, a sequence of 16 frames is generated and ROI is passed through a sequence of frames into the 3D-CNN for classification.

3.1. Activity Detection.

3.1. Activity Detection. The YOLOv4-tiny [37] object detector is a light version of the YOLOv4 [38], enhancing the detection speed. With this light version, YOLOv4 can attain around 370 frames per second (FPS) with very good accuracy on a GPU-enabled machine having a 1080Ti GPU. The YOLOv4-tiny includes cross-stage partial connections (CSP) Darket53-tiny as the backbone feature extractor instead of CSPDarket53, which was used in the original YOLOv4 [38]. The YOLOv4-tiny network uses a cross-stage partial block as a residual block, enhancing the accuracy but increasing the model complexity and eventually decreasing the FPS rate. A tradeoff is to proceed with object detection in real time on embedded devices with better accuracy. Therefore, an improved version of YOLOv4-tiny is proposed.

Figure 1 exhibits the enhanced Residual block (ResBlock instead of two CSPBlock as in YOLOv4-tiny to improve processing speed. The Enhanced ResBlock unit uses two direct path networks to handle the input representation map. In this two-path network, the path T network has three 1×1 and 3×3 convolutional (Conv) layers with stride 2, followed by another 1×1 Conv layer. Another network, Path B, has two 3×3 max pooling with stride 3 followed by a 1 x Conv layer. Compared to CSPBlock used on the original YOLOv4-tiny [37], the proposed ResBlock removes the first 3×3 Conv in CSPBlock and replaces the consequent 3×3 Conv layers with 1×1 Conv layers in the Path T network to make the detection network efficient as exhibited in Figure 1. The proposed ResBlock unit adds pooling and Conv in the Path B network. Still, this extra computation overhead is minimal as compared to reduce in computation in the Path T network. The floating-point operations (FLOPs) are analyzed to determine the computational complexity of the CSPBlock [37] and the proposed ResBlock. FLOPs can be described as follows:

$$FLOPs = \sum_{l=1}^{S} M_i^2 . F_i^2 . C_{i-1} . C_i . \tag{1}$$

Here, S is the sum of all the Conv layers, M_i^2 is the output feature vector of the corresponding lth layer, F_i^2 is the filter size, while C_1 and C_{i-1} refer to output and input channel count, respectively. For comparison, suppose an input of 224×244 with 64 channels, and using (1), FLOPs of ResBlock are used in the proposed detection model as shown in calculations in equations (2) and (3)

$$\begin{aligned} FLOPs &= 104^2 * 1^2 * 64 * 32 + 52^2 * 3^2 * 32^2 \\ &\quad + 52^2 * 1^2 * 32 * 64 + 64 * 52^2 * 2^2 + 52^2 * 1^2 * 64^2, \qquad (2) \\ FLOPs &= 6.4x10^7. \end{aligned}$$

The FLOPs of CSPBlock are used in YOLOv4-tiny against the same image

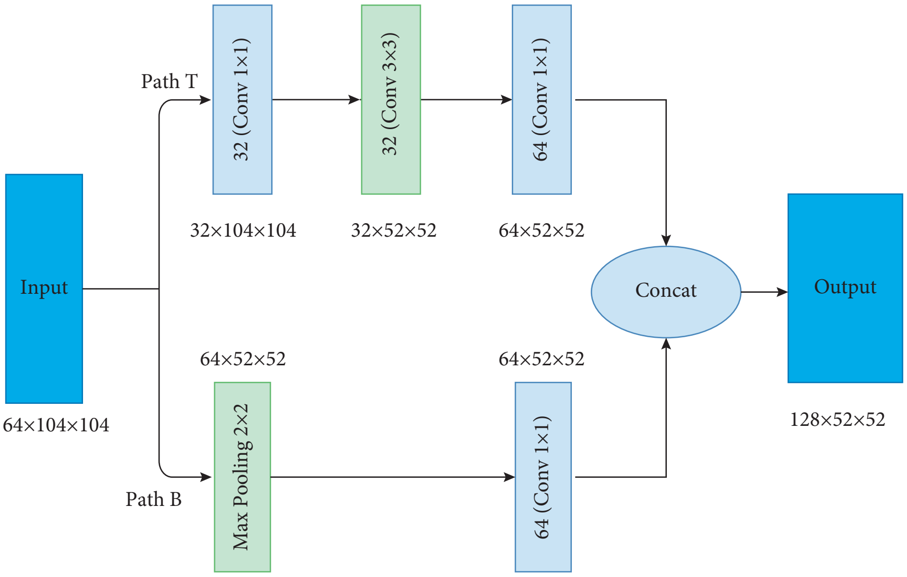

Figure 1: Enhanced ResBlock-D modules.

$$
\begin{aligned}
FLOPs &= 104^2 * 3^2 * 64^2 + 104^2 * 3^2 * 64 * 32 \\
&\quad + 104^2 * 3^2 * 32^2 + 104^2 * 1^2 * 64^2, \\
FLOPs &= 7.4x10^8.
\end{aligned}
\tag{3}
$$

From equations (2) and (3), we determine that 1 : 10 is the computation of FLOPs in ResBlock and CSPBlock. FLOPs comparison shows that ResBlock is much less complex than CSPBlock.

Although the inclusion of ResBlock in the YOLOv4-tiny detector makes it much faster than CSPBlock, it affects object detection accuracy. Therefore, two auxiliary residual blocks are also built and included in the ResBlock unit to get a better tradeoff between efficiency and accuracy. The proposed backbone network is shown in Figure 2.

The output representation of ResBlock is fused with a shallow representation of the backbone model through an element-wise sum operation. This fused representation is used as input to successive layers of the backbone model. The fusing process of representation of ResBlock and the backbone model can be expressed as

$$O^i = f^i\left(O^{i-1}\right) + Or^i. \tag{4}$$

Here in equation (4), i is the index of the layers, f^i is the fusion function between the input and output in the ith layer network, O^{i-1} refers to the i-1th layer's output and the i th layer's input, and Or^i is the output of the proposed ResBlock. This fusion catalyzes the convergence between deep and shallow networks. Moreover, with the fusion mechanism, the network learns more information to enhance the accuracy while preventing the large step-sized calculation increase.

In the backbone of YOLOv4-tiny [37], the Residual network module uses 3×3 filters for feature extraction. Although 3×3 receptive fields can extract more localized information while losing global contextual information and eventually reduces the detection accuracy. We have compensated for this loss of global representations by using two consecutive 3 x Conv layers to get the receptive field of size 5×5 in the auxiliary ResBlock. This auxiliary model passes on the obtained global representation to the backbone network. Then the backbone network joins the local contextual information extracted from the smaller (3×3) receptive field and global representation extracted from the bigger (5×5) receptive field that gives extra information about the object. This combining of global and local information not only enhances the network depth but also advances the semantic of information. The attention mechanism can process and transmit the crucial feature and eliminate the invalid features through channel suppression. We have introduced spatial and channel attention modules in the auxiliary network to extract more effective feature representations. The channel attention module emphasizes the interpretation of the informative part of the given input image and sees its meaning in it. The spatial attention module emphasizes the spatial location of the informative part of the input, supportive of channel attention. We have used the Convolutional Block Attention Module (CBAM) [18]. The used CBAM can be described as

$$F^c = M^c\left(F^i\right) \odot F^i, \tag{5}$$

$$F^s = M^s\left(F^c\right) \odot F^c. \tag{6}$$

Here in equations (5) and (6), $F^i \in R^{\mathrm{C \times H \times W}}$ the input feature map, " $\odot$ " refers to element-wise multiple, F^c and F^s are the output feature maps, M^c and M^s are the channel and spatial attention functions, respectively. The channel attention function $M^c\left(F^i\right)$ and spatial attention function $M^s\left(F^c\right)$ are expressed as

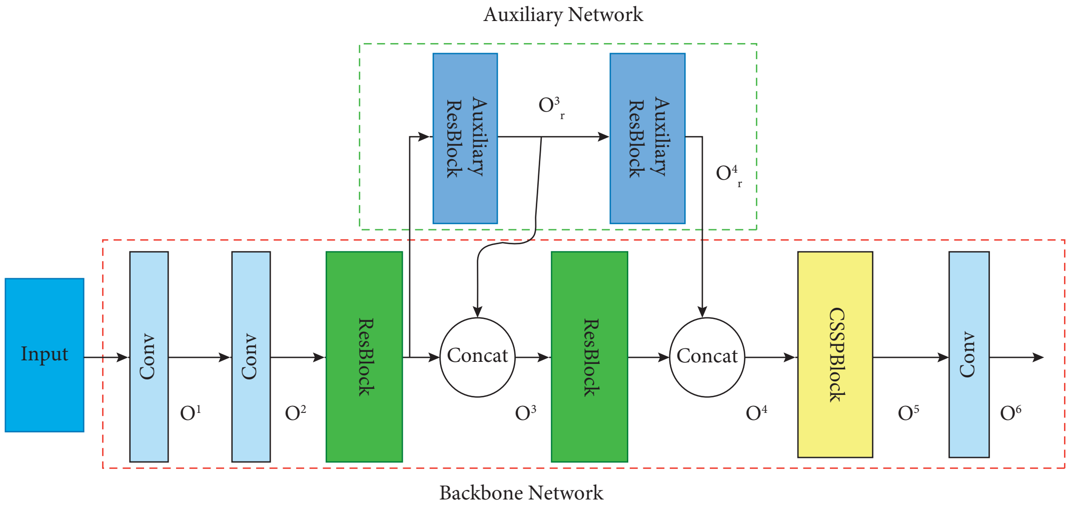

Figure 2: Proposed backbone network.

$$M^c\left(F^i\right) = \mathbf{S}\left(MLP\left(avgPooling\left(F^i\right)\right) + MLP\left(\mathrm{maxPooling}\left(F^i\right)\right)\right.$$
$$M^s\left(F^c\right) = \mathbf{C}^{5\times 5}\left[\mathrm{maxPooling}\left(F^c\right); +\mathrm{avgPooling}\left(F^c\right)\right]. \tag{7}$$

In equation (7), **S** is the sigmoid function, MLP () is the multilayer perceptron, and $\mathbf{C}^{7\times 7}$ is the convolutional operation having a filter size of 5 × 5. Max Pooling and average pooling operations in spatial attention function are combined through concatenation, referred to as ":".

Figure 3 shows the proposed auxiliary network having two convolution layers to obtain the global contextual information and channel and spatial attention to get more effective information. The output representation of the first convolution layer output received from spatial attention operating is concatenated to combine both outputs, the output of the auxiliary network. Then the final output of the auxiliary network is combined with the output of the residual network of the backbone network and used as input for the next residual network there. This joining of both outputs enhances the backbone network to extract local and global information about the object and increases the accuracy of the detection network.

The architecture of the whole YOLOv4-tiny object detector is shown in Figure 4, where the proposed network is distinguished by the blue color. Compared to YOLOv4-tiny [37], the proposed object detector has replaced both CSPBlock units with two ResBlock. Moreover, the auxiliary network is also designed using two 3 × 3 Conv layers, a channel attention module and a spatial attention module, and a concatenation operation to obtain global information. Finally, auxiliary and backbone networks are combined to make a feature extractor.

3.2. Activity Recognition. For the activity recognition in the videos, we propose a 3D CNN where we use three-dimensional convolutions to count features in both the temporal and spatial dimensions in the later stages of CNNs. Convoluting a three-dimensional kernel to the cube obtained by assembling several spatiotemporal patches in a contiguous manner yields the 3D convolution as shown in Figure 5. The feature maps in the convolution layer are connected to multiple frames arranged consecutively in the previous layer to capture motion-related details [39]. If the kernel weights are duplicated around the patch cube, the 3D convolution kernel can only select one form of a function from the patch cuboid. The number of feature maps expands as the number of layers increases on CNN, which helps create various sorts of features from the lowest available maps.

To build the 3D cube, convolve a 3D filter kernel by stacking multiple contiguous frames. The function maps are linked to multiple adjacent frames using this operation. The working mechanism of 3d CNN is described in (1), where the value at position (a, *b*, c) in the kth feature map in the lth layer is described as

$$\mathbf{v}_{\mathbf{k,l}}^{\mathbf{a,b,c}} = \mathbf{tanh}\left(\mathbf{y}_{\mathbf{kl}} + \sum_{\mathbf{m}} \sum_{\mathbf{x}=0}^{\mathrm{X}_{\mathrm{k}}-1} \sum_{\mathbf{y}=0}^{\mathrm{Y}_{\mathrm{k}}-1} \sum_{\mathbf{z}=0}^{\mathrm{Z}_{\mathrm{k}}-1} \mathbf{w}_{\mathbf{klm}}^{\mathbf{xyz}} \mathbf{v}_{(\mathbf{k}-1)\mathbf{m}}^{(\mathbf{a}+\mathbf{x})(\mathbf{b}+\mathbf{y})(\mathbf{c}+\mathbf{z})}\right). \tag{8}$$

where $\mathbf{w}_{\mathbf{klm}}^{\mathbf{xyz}}$ is the feature map linked to the m^{th} value of the kernel in the previous layer, and ZK is the 3D filter kernel size along the temporal axis. The architecture of the proposed model is shown in Figure 6.

To increase the model's efficiency, it uses 3 layers, including convolutional, pooling, and fully connected layers. In the convolutional layer, a filter layer of learned parameters converts images into processable data in this layer. Each kernel filters for a different function, and each analysis employs several kernels. In a convolution, only small parts of an image are looked at. They are assigned and transformed to an activation map that represents the image layers based on how likely it is that they belong to a certain filter class. To create three-dimensional activation maps, the kernels in a 3D CNN traverse across the three data dimensions of height, length, and depth. In pooling layers. The activation maps created during convolution are pooled or down-sampled. Pooling is similar to convolution in that it involves moving a filter around an activation map and testing a small segment simultaneously. This filter abstracts either the scanned area's

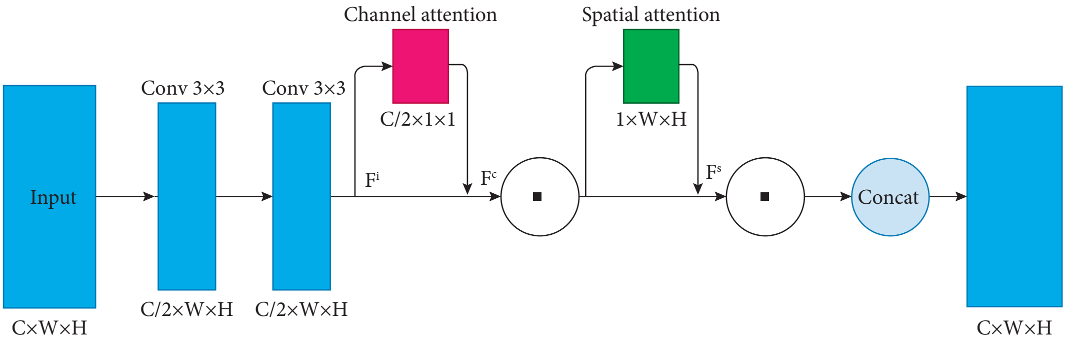

Figure 3: Auxiliary residual network.

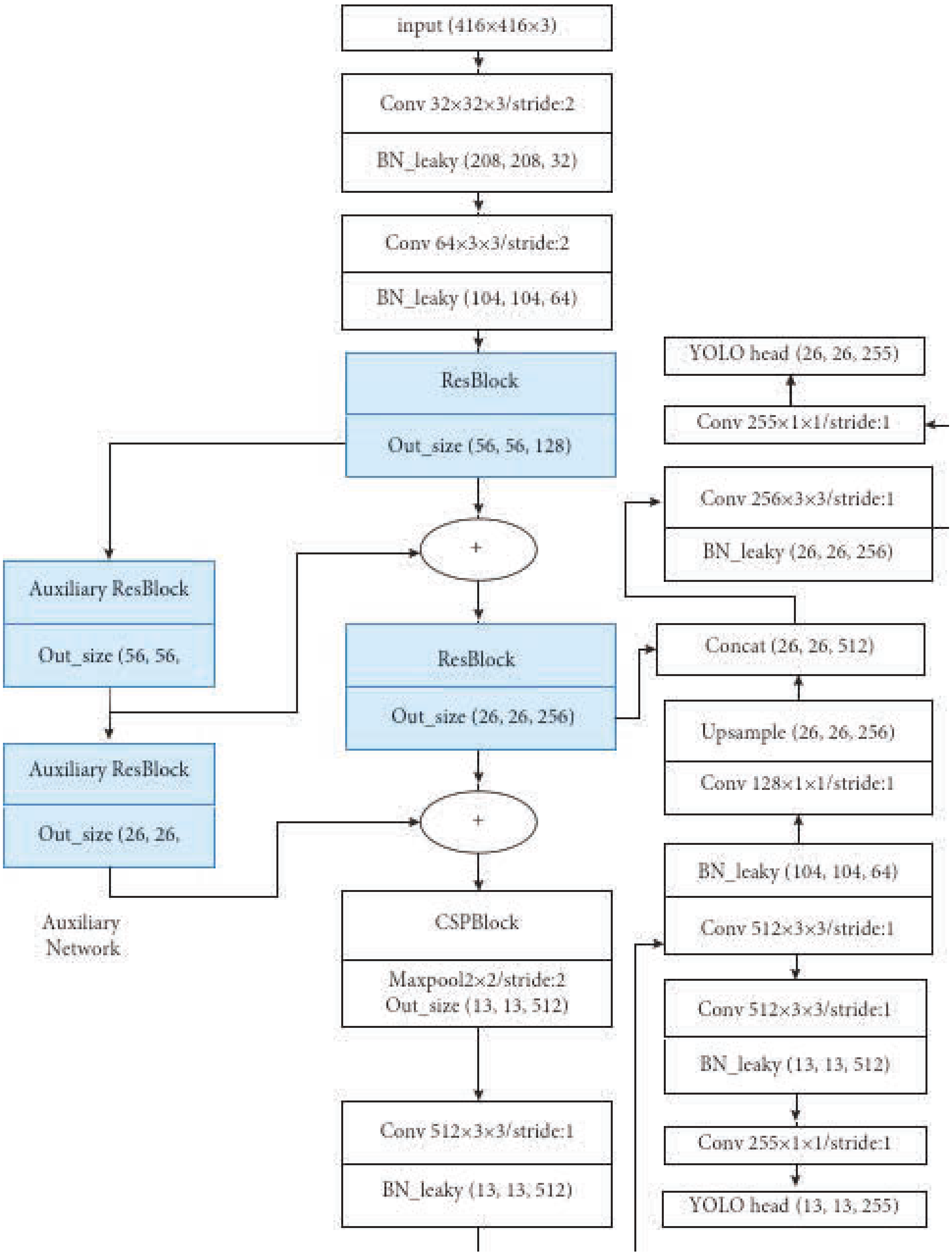

Figure 4: YOLOv4-tiny architecture with proposed changes in blue color.

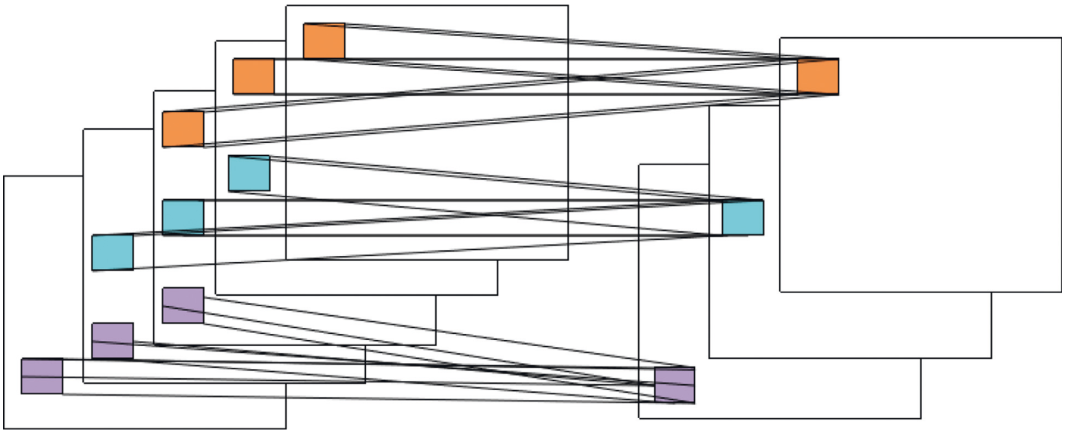

Figure 5: 3D-CNN architecture.

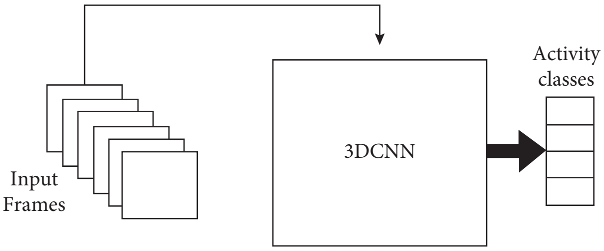

Figure 6: Flow of video recognition.

average, a weighted average dependent on the central pixel, or the extreme value of a new map.

The output layers are compressed, the probabilities found are evaluated, and the output is allotted a value, a logit, after several iterations, often thousands, of convolution and pooling. This analysis is carried out by the completely connected layer, in which each flattened output layer is interpreted by linked nodes, similar to a fully connected neural network as exhibited in Figure 6. Using hyper-parameters such as zero-padding (P), receptive field (R), stride length (S), and volume dimension (depth × width × height), calculate the spatial size of the 3D CNN output volume. We used image input of dimension $I \times J \times K$, where $I = 224$, $J = 224$, and $K = 3$ where J stands for row pixel values, I for column pixel values, and K stands for the number of channels in this work, which is three. To measure the neurons in the Convolutional layer, multiply $((W– F + 16.P)/S) + 1$. The input layer is $((224–11 + 16.0)/1) + 1 = 229$, resulting in an output volume of $229 \times 229 \times 32$, where height, width $= 224$ is the input frame's height and width, $F = 11 \times 11 \times 16$ is the 3D filter depth, $P = 0$ is the zero-padding, and $S = 1$ is the stride that leads to the output.

3.3. Smart Surveillance Using Internet of Things. The Internet of Things (IoT) has been utilized for efficient decision-making in real time. We utilized Ethernet communication to create the local server such that we have assigned a specific I.P. (Internet Protocol) address to access each camera feed present in the particular location. The flow of IoT architecture in the proposed architecture has been depicted in Figure 7. Ethernet provides minimal latency in the IoT environment and LAN (Local Area Network) for smooth communication of interconnected devices. Ethernet cables are not like other wires. The stream of any particular camera is then passed to the centralized GPU (Graphic Processing Unit) for processing and making predictions. All the decisions based on the predictions are then made available to the local network for efficient and quick response. We utilized the IoT concept because we could control each process remotely and monitor it as well. We can use different communications protocols on that according to system needs as well. The Ethernet communication protocols use this architecture to control feed monitoring and prediction remotely. The proposed work utilized the Local Area Network (LAN) technology that connects Internet devices using wired communication. It described how data is shared through a physical medium from one device to another network device. It is a link-layer protocol in the TCP/IP stack. It is based on the IEEE 802.3 standard [36]. In the proposed work, Ethernet is used to connect stationary or fixed IoT devices within an IoT system. Ethernet cable served as a wired medium for connecting computers, IP cameras, servers, switches, and routers.

We managed a stream of CCTV (Close Circuit Cameras) using a Network Video Recorder (NVR) and BNC (Bayonet Neill–Concelman) cable. The CCTV framework is arranged to impart its signal to an advanced video recorder, i.e., a DVR using BNC cable. The NVR contains five hard discs of 1 terabyte (TB) each for video film recording. It underpins HDMI (High-Definition Multimedia Interface) or VGA (Video Graphics Array) video yield, which permits focal observation on the LCD screen or TV. The proposed DVR includes video, live web-based streaming, and playback. An I.P surveillance camera communicates its signals alongside its network. I.P. security cameras used in the proposed system utilized a CAT-6 link to convey signals to the network video recorder (NVR). As a result, we achieved a higher resolution stream, efficient, real-time accessibility, and video/sound secured transmission using an IoT-based architecture.

In this work, cameras' live recordings are accessed through each specific I.P. address that is further processed by a centralized GPU-based server. The analysis and anomaly predictions are performed using the proposed computer vision-

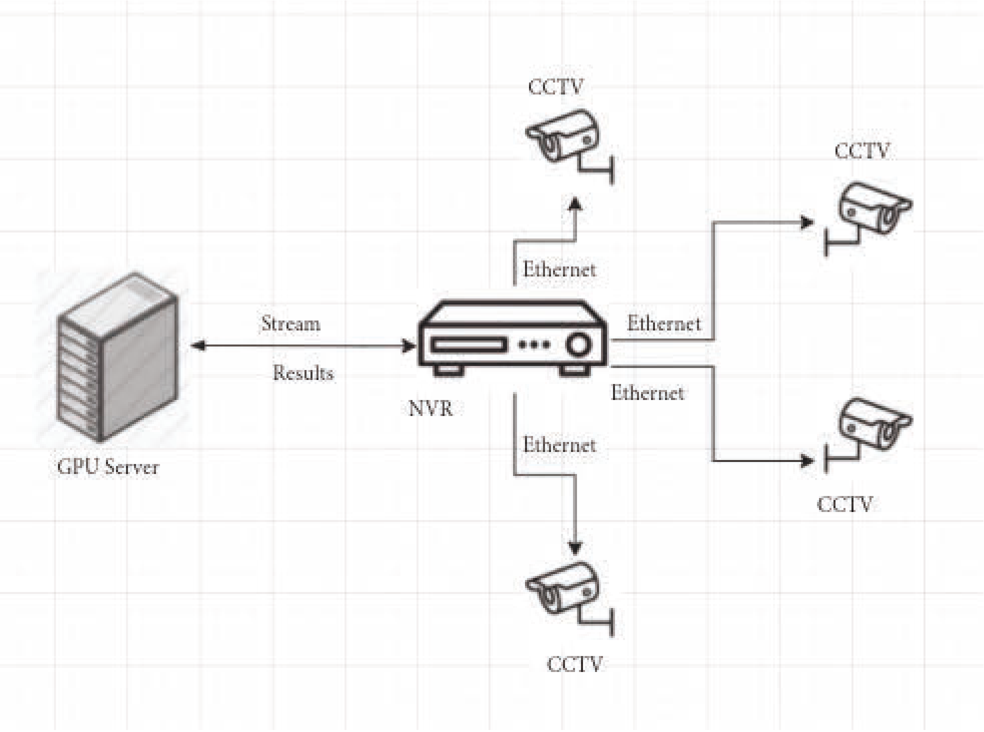

Figure 7: Internet of Things-based architecture for decision making.

based hybrid models (YOLO-v4 and 3D-CNN) to identify the specific activities in the live recordings. In case of an emergency or any suspicious activity, it can send alerts and notifications to the relevant person or authority for immediate action.

4. Experimental Results and Performance Analysis

4.1. Datasets. Benchmark datasets play a vital role in results and performance analysis in the state of the art [14]. Surveillance videos can capture a variety of real anomalies. The proposed methodology is evaluated on two major datasets. First, the UCF-Crime dataset [40] consists of various real-world anomalies, including smoke fighting, robbery, snatching, and vandalism, on which the proposed model is evaluated. These acidities are selected because they are considered prohibited. These activities are recognized based on the overall activity of the given sequence rather than the individual activities of the actors. Each mentioned activity has approximately 7000 frames in the used dataset. For training and evaluation, around 2000 frames of each activity are selected. A brief description of each activity chosen is given:

(a) Smoking: this event contains videos showing people smoking in public places such as university campuses.

(b) Fighting: this activity is based on fighting between or among people in public places such as university campuses.

(c) Snatching: this activity is based on various objects snatching, including purses, handbags, cell phones, and laptops.

(d) Gun pointing: in addition to fighting, this activity class requires gun objects in the sequence.

(e) Vandalism: this class represents a group action involving deliberate destruction of or damage to objects like buildings, vehicles, furniture, etc.

Moreover, we gathered a dataset of suspicious objects involved in those activities. We used images from the UCF crime dataset [40] and the COCO dataset [41]. Additionally, we performed preprocessing and annotation labeling on the data collected from datasets.

We have pretrained the Modified-YOLOv4, 3D-CNN network on COCO, and the whole network is fine-tuned on the CUF crime dataset, which has approximately 2000 frames for each of the five activities. The Modified-YOLOv4 is trained on the 2000 frames of each activity for 1500 epochs, while the 3D-CNN model is trained for 2000 epochs such that 80% of the dataset is devoted to training and 20% to validation. Figure 8 shows the Modified-YOLOv4 accuracy graph, from the discrepancy between training and validation accuracy. It can be deduced that the model is somewhat overfitting training data, a characteristic of deep learning models. The Modified-YOLOv4 has a training accuracy of 96.2% and a validation accuracy of 94.21%. After 1500 epochs, the training loss for Modified-YOLOv4 decreased from 8.6 to 0.19, while the validation loss decreased from 8.7 to 0.25. Figure 9 depicts the loss progression of Modified-YOLOv4 throughout training.

Figure 10 demonstrates that the 3D CNN module was similarly susceptible to overfitting since there was a discrepancy between training and validation accuracy, with validation accuracy remaining lower than training accuracy.

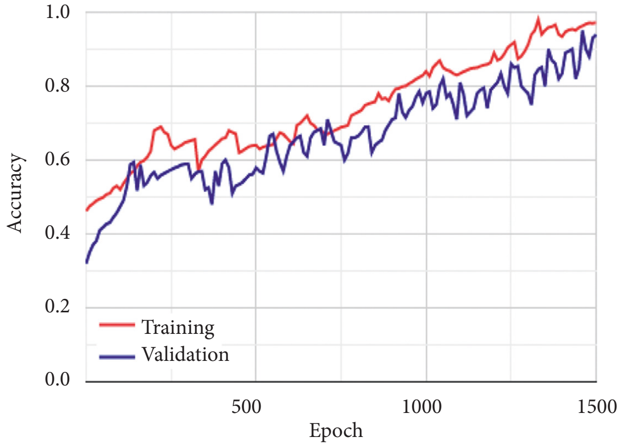

Figure 8: Accuracy graph of Modified-YOLOv4.

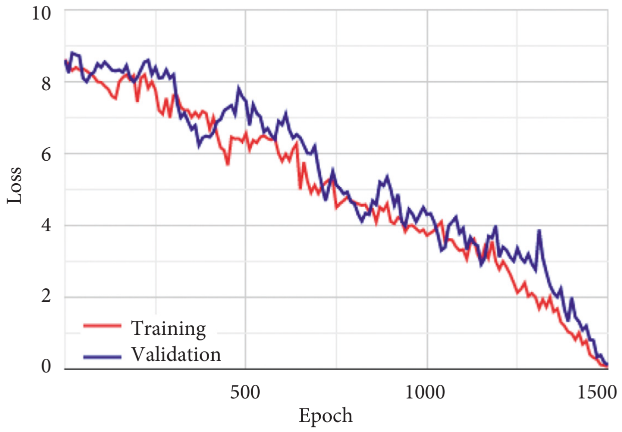

Figure 9: Loss graph of Modified-YOLOv4.

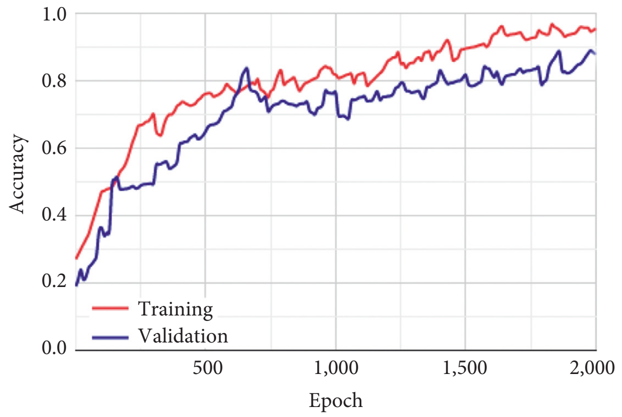

Figure 10: Training and validation accuracy graph of 3D CNN.

At the conclusion of the previous period, training accuracy was 94.8% and validation accuracy was 89.0%. Training loss (Figure 11) began decreasing from around 9.2 to 0.11, whereas validation loss began at 9.8 and finished at 0.22 in the final epoch. At some epochs, validation loss was less than training loss, but it remained significant for most of the training period.

Table 1 shows the aggregated confusion matrix of the activity recognition network. We have achieved 93.2% accuracy, 91.01% precision, and 90.1% recall. The proposed activity recognition model performed slightly poorly on the fighting and vandalism activities, while performance was highest on the gun pointing and smoking. Both precision and recall are lower because a few of the activities are

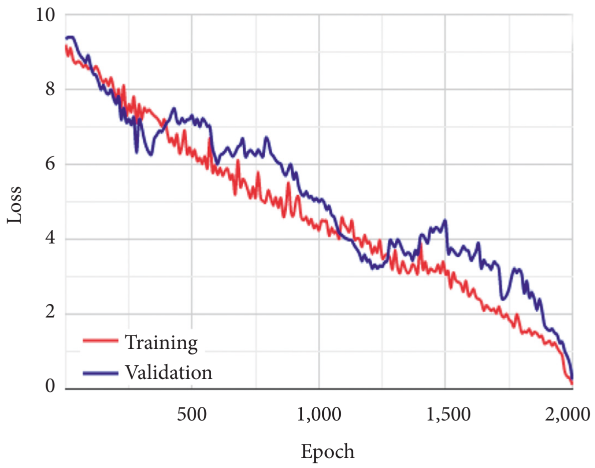

Figure 11: Training and validation Loss graph of 3D CNN.

Table 1: Confusion Matrix on proposed activities.

		Actual activity				
		Smoking	Gun pointing	Snatching	Fighting	Vandalism
Predicted activity	Smoking	48	1	1	—	—
	Gun pointing	1	49	—	—	—
	Snatching	—	1	47	2	—
	Fighting	—	—	3	44	4
	Vandalism	—	—	—	5	45

confusing. 50 activities in each category are used for this confusion matrix.

5. Conclusion

Human suspicious activity recognition is a challenging task with vast applications in video surveillance, intelligent transport systems, entertainment, and anomaly detection. Whether the event has already occurred or not, searching for the desired event through recordings is extremely time-consuming. However, a system that automatically senses any irregular or abnormal condition in advance and alerts the appropriate authorities is more appealing, and it can be used in both indoor and outdoor settings. Automatic video surveillance has tackled this issue. It is impossible to monitor CCTV events manually. Many researchers have worked on spatial information with temporal sequences for human activity recognition and detection. However, they failed to achieve impressive results in real-time. This paper presents a hybrid model which first detects the area of interest using the YOLO-v4 architecture where an anomaly or unusual activity is happening and then passes it to the 3D-CNN architecture for activity recognition based on the temporal information. The experiments performed on benchmark datasets and the 94.21% accuracy attained show the significance and robustness of the proposed architecture.

Furthermore, an IoT-based architecture has been utilized for real-time processing and efficient decision-making. The proposed multimodal is customizable, flexible, and extendable. Therefore, the system can quickly adopt new activities such as pose points, hand tracking, etc.

Data Availability

The data used in this paper are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is financially supported by Prince Mohammad bin Fahd Center for Futuristic Studies at the Prince Mohammad bin Fahd University and the World Futures Studies Federation. The authors are thankful for the support.

References

[1] T. Saba, A. Rehman, R. Latif, S. M. Fati, M. Raza, and M. Sharif, “Suspicious activity recognition using proposed deep L4-branched-ActionNet with entropy coded ant colony system optimization,” *IEEE Access*, vol. 9, pp. 89181–89197, 2021.

[2] A. R. Khan, T. Saba, M. Z. Khan, S. M. Fati, and M. U. G. Khan, “Classification of human’s activities from gesture recognition in live videos using deep learning,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 10, Article ID e6825, 2022.

[3] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, “Anomaly-based intrusion detection system for IoT networks

through deep learning model," *Computers & Electrical Engineering*, vol. 99, Article ID 107810, 2022.

[4] I. Imran, S. Din, G. Jeon, and G. Fortino, "Towards collaborative robotics in top view surveillance: a framework on using deep learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1253–1270, 2021.

[5] M. A. Khan, H. Arshad, R. Damaševičius et al., "Human Gait Analysis: A Sequential Framework of Lightweight Deep Learning and Improved Moth-Flame Optimization Algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[6] Y. Al-Hamar, H. Kolivand, M. Tajdini, T. Saba, and V. Ramachandran, "Enterprise credential spear-phishing attack detection," *Computers & Electrical Engineering*, vol. 94, Article ID 107363, 2021.

[7] H. Yar, T. Hussain, Z. A. Khan, D. Koundal, M. Y. Lee, and S. W. Baik, "Vision sensor-based real-time fire detection in resource-constrained IoT environments," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–15, 2021.

[8] F. Orujov, R. Maskeliūnas, R. Damaševičius, W. Wei, and Y. Li, "Smartphone based intelligent indoor positioning using fuzzy logic," *Future Generation Computer Systems*, vol. 89, pp. 335–348, 2018.

[9] I. Abunadi, "Enterprise architecture best practices in large corporations," *Information*, vol. 10, no. 10, p. 293, 2019.

[10] B. Al, F. Orujov, R. Maskeliūnas, R. Damaševičius, and A. Venčkauskas, "Fuzzy logic type-2 based wireless indoor localization system for navigation of visually impaired people in buildings," *Sensors*, vol. 19, no. 9, p. 2114, 2019.

[11] G. Vallathan, A. John, C. Thirumalai, S. Mohan, G. Srivastava, and J. C. W. Lin, "Suspicious activity detection using deep learning in secure assisted living IoT environments," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3242–3260, 2021.

[12] M. Yousuf, Z. Mehmood, H. A. Habib, T. Mahmood, and M. Rehman, "A novel technique based on visual words fusion analysis of sparse features for effective content-based image retrieval," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–13, 2018.

[13] I. M. Nasir, M. Raza, J. H. Shah, S. H. Wang, U. Tariq, and M. A. Khan, "HAREDNet: a deep learning based architecture for autonomous video surveillance by recognizing human actions," *Computers & Electrical Engineering*, vol. 99, Article ID 107805, 2022.

[14] J. L. González, C. Zaccaro, J. A. García, L. M. Morillo, and F. Caparrini, "Real-time gun detection in CCTV: an open problem," *Neural Networks*, vol. 132, pp. 297–308, 2020.

[15] C. D. J. I. Zong, "Smart security system for suspicious activity detection in volatile areas," *Journal of Information Technology and Digital World*, vol. 02, no. 01, pp. 64–72, 2020.

[16] H. Kolivand, M. S. Rahim, M. S. Sunar, A. Z. A. Fata, and C. Wren, "An integration of enhanced social force and crowd control models for high-density crowd simulation," *Neural Computing & Applications*, vol. 33, no. 11, pp. 6095–6117, 2021.

[17] M. E. Issa, A. M. Helmi, M. A. A. Al-Qaness, A. Dahou, M. A. Elaziz, and R. Damaševičius, "Human activity recognition based on embedded sensor data fusion for the internet of healthcare things," *Healthcare*, vol. 10, no. 6, p. 1084, 2022.

[18] G. Şengül, E. Ozcelik, S. Misra, R. Damaševičius, and R. Maskeliūnas, "Fusion of smartphone sensor data for classification of daily user activities," *Multimedia Tools and Applications*, vol. 80, no. 24, pp. 33527–33546, 2021.

[19] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon detection in real-time CCTV videos using deep learning," *IEEE Access*, vol. 9, pp. 34366–34382, 2021.

[20] F. Afza, M. A. Khan, M. Sharif et al., "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image and Vision Computing*, vol. 106, Article ID 104090, 2021.

[21] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: a survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, 2008.

[22] Y. ZhuZhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware activity modeling using hierarchical conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1360–1372, 2015, Jul.

[23] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694–4702, Boston, MA, USA, June 2015.

[24] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980, Las Vegas, NV, USA, June 2016.

[25] P. Khaire, P. Kumar, and J. Imran, "Combining CNN streams of RGB-D and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018.

[26] M. Gnouma, A. Ladjailia, R. Ejbali, and M. Zaied, "Stacked sparse autoencoder and history of binary motion image for human activity recognition," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2157–2179, 2019.

[27] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Y. Wang, "Driver activity recognition for intelligent vehicles: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5379–5390, 2019.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[29] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, June 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016.

[31] Z. Chang, X. Ban, Q. Shen, and J. Guo, "Research on three-dimensional motion history image model and extreme learning machine for human body movement trajectory recognition," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–15, 2015.

[32] X. F. Ji, Q. Q. Wu, Z. J. Ju, and Y. Y. Wang, "Study of human action recognition based on improved spatio-temporal features," *International Journal of Automation and Computing*, vol. 11, no. 5, pp. 500–509, 2015.

[33] Y. Zhong, B. Xu, G. T. Zhou, L. Bornn, and G. Mori, "Time Perception Machine: Temporal point Processes for the when, where and what of Activity Prediction," 2018, https://arxiv.org/abs/1808.04063.

[34] T. Saba, “Intrusion detection in smart city hospitals using ensemble classifiers,” in *Proceedings of the 2020 13th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 418–422, IEEE, Liverpool, United Kingdom, June 2020.

[35] J. Patalas-Maliszewska, D. Halikowski, and R. Damaševičius, “An automated recognition of work activity in industrial manufacturing using convolutional neural networks,” *Electronics*, vol. 10, no. 23, p. 2946, 2021.

[36] D. A. John, “IEEE 802 LMSC,” 2022, https://standards.ieee.org/standard/802_3-2018.html.

[37] A. Bochkovskiy, “Darknet: Open-Source Neural Networks in Python,” 2021, https://github.com/AlexeyAB/darknet.

[38] A. Bochkovskiy, C. Y. Wang, and H. Y. Liao, “Yolov4: Optimal Speed and Accuracy of Object Detection,” 2020, https://arxiv.org/abs/2004.10934.

[39] J. Arunnehru, G. Chamundeeswari, and S. P. Bharathi, “Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos,” *Procedia Computer Science*, vol. 133, pp. 471–477, 2018.

[40] W. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” 2018, https://arxiv.org/abs/1801.04264.

[41] T. Y Lin, M. Michael, B. Serge et al., “Microsoft coco: common objects in context,” 2014, https://arxiv.org/abs/1405.0312.