Computers, Materials & Continua
DOI:10.32604/cmc.2022.018181
Article

Tech Science Press

Anomaly Based Camera Prioritization in Large Scale Surveillance Networks

Altaf Hussain[1,2], Khan Muhammad[1], Hayat Ullah[1], Amin Ullah[1,4], Ali Shariq Imran[3], Mi Young Lee[1], Seungmin Rho[1] and Muhammad Sajjad[2,3,*]

[1]Department of Software, Sejong University, 143-747, Seoul, Korea
[2]Digital Image Processing Lab, Islamia College Peshawar, 25000, Pakistan
[3]Color Lab, Department of Computer Science, Norwegian University of Science and Technology (NTNU), 2815, Gjøvik, Norway
[4]CORIS Institute, Oregon State University, 97331, Oregon, USA
[*]Corresponding Author: Muhammad Sajjad. Email: muhammad.sajjad@icp.edu.pk
Received: 28 February 2021; Accepted: 06 May 2021

Abstract: Digital surveillance systems are ubiquitous and continuously generate massive amounts of data, and manual monitoring is required in order to recognise human activities in public areas. Intelligent surveillance systems that can automatically identify normal and abnormal activities are highly desirable, as these would allow for efficient monitoring by selecting only those camera feeds in which abnormal activities are occurring. This paper proposes an energy-efficient camera prioritisation framework that intelligently adjusts the priority of cameras in a vast surveillance network using feedback from the activity recognition system. The proposed system addresses the limitations of existing manual monitoring surveillance systems using a three-step framework. In the first step, the salient frames are selected from the online video stream using a frame differencing method. A lightweight 3D convolutional neural network (3DCNN) architecture is applied to extract spatio-temporal features from the salient frames in the second step. Finally, the probabilities predicted by the 3DCNN network and the metadata of the cameras are processed using a linear threshold gate sigmoid mechanism to control the priority of the camera. The proposed system performs well compared to state-of-the-art violent activity recognition methods in terms of efficient camera prioritisation in large-scale surveillance networks. Comprehensive experiments and an evaluation of activity recognition and camera prioritisation showed that our approach achieved an accuracy of 98% with an F1-score of 0.97 on the Hockey Fight dataset, and an accuracy of 99% with an F1-score of 0.98 on the Violent Crowd dataset.

Keywords: Camera prioritisation; surveillance networks; convolutional neural network; computer vision; deep learning; resource-constrained device; violent activity recognition

1 Introduction

Following the development of computer vision and pattern recognition technology, video surveillance systems have been widely deployed, and their functionality is improving rapidly with the aim of preventing emergencies and crimes [1]. Current systems mainly rely on traditional approaches that are time-consuming, laborious, and prone to misdetection of violent activities due to need for round-the-clock monitoring of large numbers of camera nodes. Video surveillance, and particularly human activity recognition systems, have strong prevention capabilities due to their improved visualisation, accuracy and real-time feedback, and these are now considered essential aspects of a security system. Wireless visual sensor networks (WVSNs) have recently emerged as a new type of sensor-based intelligent system, and their performance exceeds that of existing surveillance sensor networks. Furthermore, due to their small size and the dense spatial coverage that can be achieved, WVSNs can be flexibly deployed for various computer vision tasks, including patient monitoring [2], distributed multimedia-based surveillance systems [3], military, police, airport [4], border [5], urban and transport [6] applications, and other areas of public interest. WVSN surveillance systems have an extensive range of applications worldwide, but their implementation has remained a challenge, due to the need for connectivity between multiple sensors and their complicated setup. For instance, monitoring a large area requires a large number of WVSNs. To accurately monitor an entire set of cameras in real time, WVSNs require massive amounts of human resources, computational power and bandwidth. In addition, vision-based sensors require a high bandwidth to transmit data between cameras and servers. Efficient monitoring of the target area also requires extensive computation to detect events and anomalies. A similar study reported the need for extensive computational resources due to the use of multiple sets of parameters in the proposed model [7]. The literature in the area of violent activity recognition is mainly divided into handcrafted and deep learning-based methods, and these are discussed below.

1.1 Handcrafted Feature-Based Approaches

The success of handcrafted feature-based approaches relies heavily on manually engineered feature extraction techniques. Many researchers have utilised handcrafted features for the detection of violent activities. For instance, Hassner et al. [8] introduced extractor flow vectors using a violent flow descriptor (ViF). They then used a support vector machine (SVM) to classify these features into violent and non-violent crowds. Similarly, Huang et al. [9] used SVM to analyse the statistical properties of the optical flow for violent crowd recognition. Zhang et al. [10] used a Gaussian model of the optical flow for violent region extraction, and an orientation histogram of the optical flow with linear SVM was used to classify video frames into violent and non-violent classes. Gao et al. [11] proposed an oriented violent flow descriptor (OViF) that utilised both the magnitude of motion and orientation information for the recognition of violent activity. Chen et al. [12] used spatiotemporal interest points, including space-time interest points (STIPs) [13] and a motion scale-invariant feature transform (Mo SIFT) [14], for violent activity recognition. Similarly, Lloyd et al. [15] proposed a new descriptor called a grey level co-occurrence texture measure to detect violence and abnormal activity in crowds. Fu et al. [16] analysed three attributes of motion (region, magnitude, and acceleration) for violence detection from surveillance videos. For feature extraction, the optical flow magnitude and orientation (HOMO) [17] was also used. A sliding window method was used in [18] with an improved Fisher vector (IFV) for violent activity recognition.

1.2 Deep Learning-Based Approaches

Due to the wide range of variations in pose, viewpoint and scale in visual data, accurate recognition of violent activities is complex and challenging. Researchers from the artificial intelligence and computer vision communities have recently contributed to this area by identifying human activities using deep learning-based approaches. CNNs extract features in a hierarchical way, where the initial layers learn local features and the higher layers learn global features from the visual data [19]. Although recurrent neural networks (RCNNs) and 3DCNNs are mostly used for violent activity recognition [20], an RNN is not able to extract features directly; a CNN is typically used for feature extraction purposes, and these features are then passed to an RNN for classification. A 3DCNN is an end-to-end method that is very widely used to extract spatio-temporal features for violent activity recognition [21–28]. For instance, Tran et al. [21] used a 3DCNN architecture for activity recognition in which $3 \times 3 \times 3$ filters were convolved with a sequence of 16 frames.

Similarly, Carreira et al. [22] modified 2D filters pretrained on ImageNet to create 3D versions for activity recognition. These modified filters achieved better accuracy than a filter that was randomly initialised. Stroud et al. [26] introduced a distilled 3D network (D3D) for activity recognition. Diba et al. [23] employed a temporal transition layer with DenseNet3D, in a scheme that used transfer learning from a 2DNN model. Serrano et al. [29] used Hough forests with 2DCNN to detect violent activity, and their proposed approach obtained an accuracy of 94.6% on the Hockey Fight dataset. However, 3DCNNs have high computational requirements, making them unsuitable for use in standard surveillance systems due to resource constraints. In this paper, we solve this issue by introducing a new lightweight 3DCNN model that is less computationally intensive and can easily be deployed in common CCTV camera surveillance systems.

Within such large amounts of video data, very few scenes are important in terms of allowing a machine to understand human activity. For example, theft from a shopping mall happens very rarely. When a human performs any activity, there is some sort of bodily movement, such as movements of arms and legs. In these situations, the detection of moving objects from a sequences of frames is a challenging and crucial task, and is the first step in any video analytics system such as video surveillance [30], target detection [31], human tracking [32], and robot control [33]. The selection of salient motion frames from WVSN nodes is a crucial aspect of video processing that helps us analyse only important clips, thereby effectively minimising the execution time and improving the accuracy of the violent activity recognition system. Several techniques for automatically detecting salient frames have been developed that can separate the moving objects (foreground) from the scene (background) in videos. It is difficult to accurately segment foreground objects due to variations in illumination, ghosting of the foreground aperture, and the presence of unconstrained environments. Over the years, approaches based on optical flow [34], background subtraction [35], and frame differencing [36] have been actively used to detect motion between two consecutive frames. Optical flow is used to detect the simple apparent motion of an object in two consecutive frames; however, this is computationally expensive and produces inaccurate results due to its susceptibility to noise, variations in illumination, and fake motion. The second method, frame differencing, is a straightforward and commonly used technique for identifying moving objects, but is susceptible to variations in illumination and camera jitter. Unlike optical flow, the frame difference technique is computationally efficient and is particularly used by resource-constrained devices to detect moving objects in video frames.

Existing large-scale WVSNs consist of numerous wireless camera sensors and are used to monitor suspicious human activities. However, these systems have several drawbacks, such as

inadequate recognition of salient information, streaming of all imaging data, high bandwidth and storage requirements, ineffective monitoring, and late responses to crime or abnormal activities. Other significant issues related to WVSN-based surveillance include scattered background viewpoint variation and changes in lighting. The task of camera prioritisation in large-scale WVSNs also becomes more challenging when large numbers of nodes and continuous streaming are used [37]. Researchers around the world are making efforts to tackle these challenges. For instance, Mehmood et al. [38] proposed a saliency-aware data prioritisation framework that selects semantically relevant information for each node and then transmits these data to a sink node for further processing. The main limitation of their method is that they used handcrafted features to extract salient information, an approach that gives limited performance with real-time data. To rank salient information and remove redundancy, Thomas et al. [39] used a perceptual system to detect road events. Another technique inspired by the idea of perceptual computation was a low-computation video summarisation framework for resource-constrained devices such as the Raspberry Pi [40].

The abovementioned approaches were developed in an attempt to solve several challenges such as variations in pose, viewpoint and scale, complex crowd patterns in visual data, and data prioritisation. However, there are still numerous challenges that need to be addressed. For instance, the studies in [8–17] used handcrafted approaches that were not capable of learning discriminative features from violence datasets. The authors of [20–28] proposed 3DCNN models for violent activity that performed rather well, but due to the large number of computations involved, these were not suitable for deployment in real-world surveillance systems. Similarly, the authors of [36,37] developed a surveillance system that prioritised video camera content. A surveillance system usually consists of resource-constrained devices such as CCTV cameras, and there is therefore an urgent need for a system that can accurately recognise violent activity in a complex environment with lower computational requirements. In addition, there is a need for an intelligent prioritisation technique that can select only those cameras that are transmitting violent activity from among a set of normal feeds, to allow for smart monitoring of large WVSNs, reduce the memory storage required, and utilise the resources of WVSNs more efficiently.

In this paper, we propose an energy-efficient deep learning-based framework for camera prioritisation in large-scale WVSNs. The key contributions of this work can be summarised as follows.

i. A novel camera prioritisation framework is proposed for economical hardware devices (e.g. the Raspberry Pi) based on violent activity recognition in large-scale WVSNs. Salient motion is the key feature for activity recognition. A lightweight frame differencing technique is incorporated to extract frames containing salient motion, thus ensuring the efficient utilisation of resources.
ii. Human activity consists of a sequence of motion patterns in consecutive frames, meaning that both spatial and temporal features need to be learned for this task. A novel lightweight 3DCNN architecture over resource-constrained devices is proposed, which outperforms other state-of-the-art techniques in terms of accuracy on benchmark datasets.
iii. A novel linear threshold gate sigmoid (LTGS) method is used to prioritise cameras based on violent activity in large-scale WVSNs by exploiting both the metadata and the probabilities predicted by the proposed 3DCNN model, thereby reducing the dependency on human monitoring, the energy required, and the bandwidth consumption.

The remainder of this paper is organised as follows. Section 2 presents the proposed methodology. The experimental setup and evaluation are described in Section 3, and Section 4 concludes the paper and suggests future work.

2 Proposed Method

This section introduces our novel camera prioritisation framework for efficient surveillance, in which an individual camera node from a large-scale WVSN is prioritised based on the detection of violent activity. The proposed framework is divided into three steps. In the first step, we perform motion detection, and select only motion-specific frames from the video streams captured by the WVSNs. In the second step, a sequence of 16 salient frames is forwarded to a trained 3DCNN model to extract the spatio-temporal features. The extracted features are then fed to a Softmax classifier, which generates categorical probabilities in which the class with higher probability is considered to be the predicted class. Finally, based on the predicted probabilities, our proposed framework prioritises cameras with high probability of showing violent activity, using LTGS. The overall workflow of the proposed intelligent camera prioritisation framework is illustrated in Fig. 1.

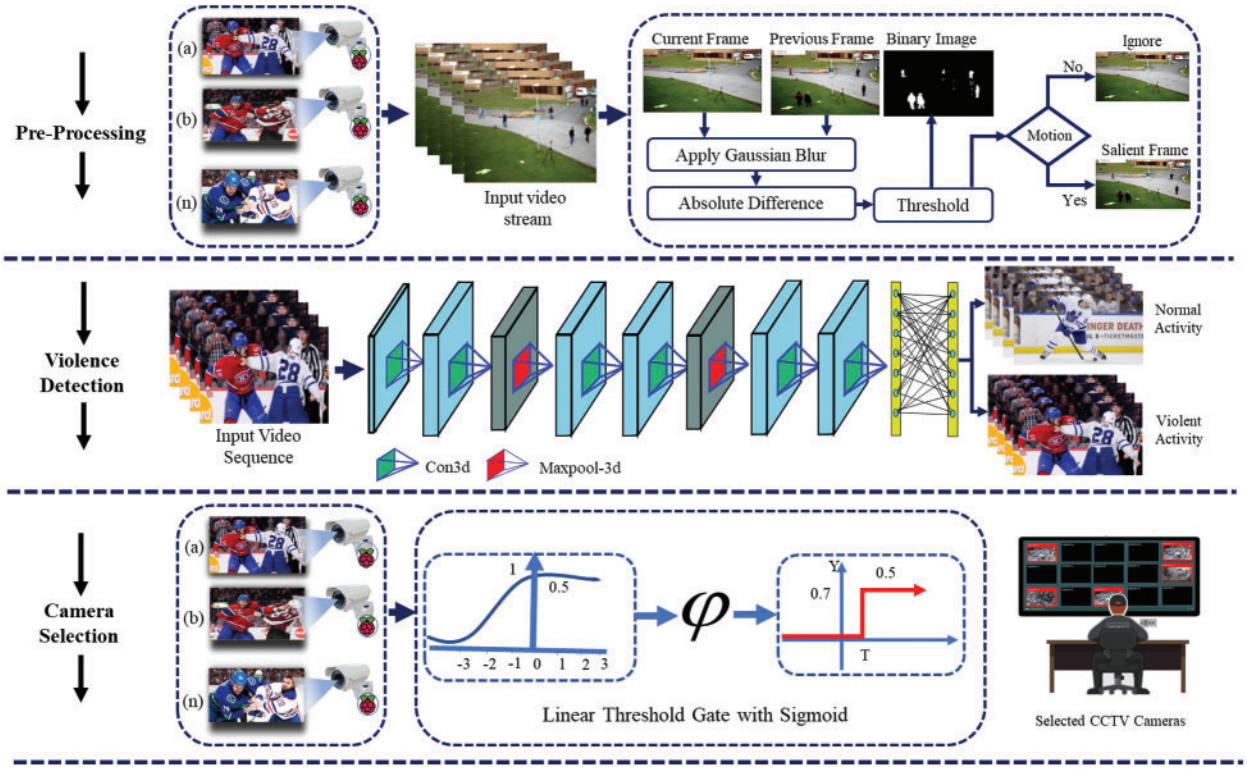

Figure 1: The proposed camera prioritisation framework for a large-scale surveillance network. In the preprocessing step, the surveillance video streams are preprocessed using the background subtraction method to extract only motion-specific frames. In the second step, a 3DCNN is utilised to extract spatio-temporal features from the sequence of frames to classify the underlying activity. Finally, based on the violent activity detected in the video stream, a specific camera is assigned with high priority

2.1 Preprocessing Phase

The frame differencing technique plays a very important role in the efficient use of resources in large-scale WVSNs, as it can help in selecting only motion-specific frames, i.e., moving cars or pedestrians. In our case, a resource-constrained Raspberry Pi device is used in the preprocessing stage to select only frames containing moving objects from the input video stream. The detection of salient objects is a difficult task due to the range of different viewpoints, occlusions, and cluttered backgrounds in the frames [17]. In WVSNs, multiple visual sensors are interconnected to enable the target area to be efficiently monitored. Processing of the video stream from each camera is not important, as it is an inefficient use of the available computational resources. There is therefore a need for a system that can select only motion-relevant frames. A lightweight frame differencing technique is applied here to identify salient motion frames, as frame differencing is the most precise and straightforward technique of detecting minor temporal changes. A pair of consecutive frames $f_i - f_{i+1}$ are smoothed and processed to remove noise; this removes high-frequency content such as noise and edges from the images. There are several different techniques for smoothing, such as averaging, median, bilateral filtering, and Gaussian blur, in which the pixels close to the centre of the filter are given more weight than those further away from the centre. We conducted experiments and concluded that due to the use of a Gaussian kernel, Gaussian blur is highly effective in removing noise from images. For the selection of salient frames, a pixel-wise absolute difference is calculated for each pair of consecutive frames $D_{image} = |f_i - f_{i+1}|$. When the value of the absolute difference is higher than a pre-defined threshold T (in our case, $T = 0.7$), these video frames are selected as salient frames and considered for further processing. The complete flow of the motion detection process is described in Algorithm 1.

Algorithm 1: Selection of salient frames

Input: Video stream

1. Take two consecutive input video frames (f_i, f_{i+1}).
2. Apply Gaussian blur on $(f_i - f_{i+1})$ to remove noise.
3. Find the pixel-wise absolute difference of each pair of consecutive frames $D_{image} = \frac{1}{N}\sum_{i=1}^{N} \left| f_{i(i)} - f_{i+1(i)} \right|$
4. **For** each frame (f_i, f_{i+1})
   IF $D_{image} <= 0.7$
   f_i and f_{i+1} are non-salient frames.
   Else
   f_i and f_{i+1} are salient frames.
   End
   End For

Output: Salient frames

2.2 Violence Detection

2DCNN architectures are widely used to extract spatial information from still images for a variety of computer vision tasks, such as image classification and image retrieval. However, the analysis of human activity in videos is challenging compared to the classification of still images, as human activity/action is encoded across multiple frames that involve both spatial and temporal information. A variety of existing methods have used a 2DCNN to extract the spatial correlations from a video, which also includes temporal information. For example, in [41,42], the

authors used a 2DCNN to process multiple frames and all the temporal features are collapsed. While in 3DCNN, a 3D filter is convolving on spatial and across the multiple frames to extract spatial and temporal information. After the 3D convolutional operation, a feature map is obtained that captures the spatial and temporal information. The feature maps are extracted from multiple frames to capture the temporal information. In our case, the feature maps are a combination of 16 consecutive frames with spatial dimensions of 112×112. The values at location x, y, z of the q^{th} extracted feature map in the p^{th} layer with bias t_{pq} are given by Eq. (1):

$$N_{pq}^{xyz} = \tanh\left(t_{pq} + \sum_{k} \sum_{a=0}^{A_p-1} \sum_{b=0}^{B_p-1} \sum_{c=0}^{C_p-1} w_{pqk}^{abc} N_{(p-1)k}^{(x+a)(y+b)(z+c)}\right) \tag{1}$$

Here, *Cp* represents the size of the 3D filter used to learn the temporal dimension, and in w_{pqk}^{abc}, the values at position (a, b, c) represent the filter convolved on the k^{th} feature map in the previous layer. The first version of 3DCNN was introduced in 2014; it was implemented in a deep learning framework called Caffe [43], and achieved state-of-the-art performance in an activity recognition task. Inspired by the exceptional performance of this network, we designed a similar 3DCNN architecture with pruning of the higher layers for the task of violent activity recognition. In our proposed 3DCNN architecture, we used eight 3D convolutional layers, two max-pooling layers, and one fully connected layer. In the final layer, a Softmax activation function is used as a binary classifier. To extract the deep spatial features from the input video stream, a $3 \times 3 \times 3$ filter is used in the convolution layer with stride one. The input shape of the model is based on the batch size, depth, rows, columns and channels, and can be represented as $16 \times 16 \times 112 \times 112 \times 3$. A leaky ReLU is used as the activation function to overcome the dying ReLU problem. In a CNN, the fully connected layer is extremely expensive in terms of computation, since each neuron is directly connected to each pixel of the input image. We therefore use a fully connected layer with 512 neurons.

2.3 Camera Selection

In large-scale WVSNs, the streaming and monitoring of gigantic amounts of data is impractical due to limited human resources. Each surveillance system consists of large numbers of sensors that require a high bandwidth to transmit their raw video streams, and these video streams involve costly computation when detecting events and anomalies. Although due to its continuous nature, a video stream poses critical problems for a human analyst in terms of identifying the important portions, it is vital to apply visual analytics at the point of data collection. To overcome the drawbacks of a traditional multi-camera WSN, we propose a salient motion detection scheme based on a prioritisation framework for visual data, in which all nodes autonomously prioritise certain visual content and the cameras showing it. The main goal of the proposed surveillance system is to focus solely on anomaly-specific cameras, not only to avoid unimportant streams but also to provide an efficient surveillance system. An overview of the proposed scheme is shown in Fig. 2. The first step involves understanding the scene by classifying the input video stream into salient and non-salient classes; the output is a predicted probability score for each frame of the streams captured by different camera nodes. These scores are then forwarded to the second step together with the weight values of the corresponding cameras.

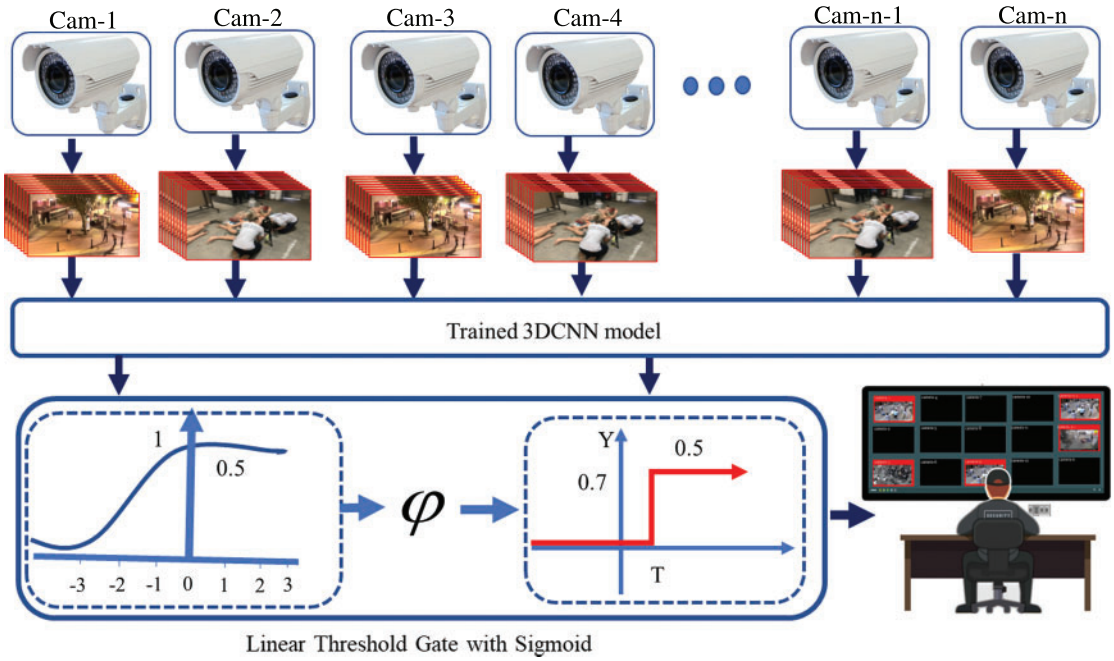

Figure 2: The proposed camera prioritisation scheme with a symbolic representation of LTGS

In the second step, the sensitivity of each camera is computed based on the input frame saliency by the LTGS module (using a sigmoid activation function), which processes two inputs: the probability of the frame under observation and the metadata (weight) value of the corresponding camera. It then generates a single value that determines the priority of the camera. Metadata are data about the camera, such as its importance and location. This is crucial, since in some locations such as banks and border control stations, we want to create a high level of surveillance by selecting camera nodes with high priority. In our experiments, we used metadata values from one to six, where a score of six means that the location of the camera is very important and one means the location is normal.

$$\varphi = \partial(w_i x_i) \tag{2}$$

where i is the number of the camera, w represents the metadata for the specific camera, x represents the predicted probability of the camera, ∂ is the sigmoid function, and ϕ indicates the priority of the camera.

$$C_F = \begin{Bmatrix} 1 & if\, \varphi \geq T \\ 0 & elsewhere \end{Bmatrix} \tag{3}$$

where c indicates the specific camera node, F represents the priority flag, and T is a threshold with a default value of 0.7. If the value of ϕ exceeds the threshold T, then the camera priority flag F is set, and this particular camera is prioritised over the others.

3 Experimental Results

In this section, we describe the details of our experiments and present a comparison of the proposed framework with other state-of-the-art techniques. To evaluate the performance of the

proposed method, we conducted various experiments on the publicly available Hockey Fight [44] and Violent Crowd [8] violence detection datasets. We examined our network with different learning rates, numbers of convolutional layers, activation functions, and other temporal parameters such as variations in the sequence size. We also compared our proposed model with different handcrafted and deep learning methods. Finally, the proposed system was quantitatively evaluated for camera prioritisation tasks, with and without a weighted strategy.

3.1 Experimental Setting

All the experiments were conducted on a Core i5 CPU equipped with NVIDIA GeForce GTX 1070 (8 GB) and 24 GB onboard memory with Windows 10 operating system, using the Tensorflow deep learning framework. For video recording and preprocessing, a Raspberry Pi model 3 was used with a 64 GB Micro SD-card, 1 GB RAM (LPDDR2 (900 MHz)), 1.2 GHz CPU (4× ARM Cortex-A53), networking (10/100 Ethernet, 2.4 GHz 802.11n wireless). The camera aperture (F) was 2.0, the focal length was 3.04 mm, the angle of view (diagonal) was 62.2°, and the image resolution was 3280×2464 (with support for 1080p30, 720p60 and 640×480p90 video recording) with a Raspbian operating system.

3.2 Datasets

This section presents a detailed overview of the datasets used in our experiments. We used two existing benchmark datasets, Hockey Fight and Violent Crowd.

3.2.1 Hockey Fight Dataset

This dataset was introduced by Nievas et al. [44] and contains a total of 1000 videos, 500 of which show violence (fights), and the remaining 500 show non-violent (normal) activities. In the violence class, all clips were collected from fights during a hockey game. The entire set of video clips was taken from a National Hockey League (NHL) game, and each video clip consists of 50 video frames with resolution $360 \times 288 \times 3$. Examples are shown in Fig. 3, and the details are given in Tab. 1. After training, we evaluated the performance of our proposed architecture on the test data. The confusion matrix for this dataset is shown in Fig. 4a.

3.2.2 Violent Crowd Dataset

The Violent Crowd dataset contains 246 videos taken from YouTube, and was introduced by Hassner et al. [8]. Originally, this dataset consisted of five different categories of violent and non-violent activities. In our experiments, we merged these categories into two classes containing violent and non-violent activity. In each category, there are 123 videos, and each clip has 50 to 150 frames with dimensions of $320 \times 240 \times 3$. An example frame from the Violent Crowd dataset is shown in Fig. 3, and details of the dataset are given in Tab. 1. The experimental results obtained from the Violent Crowd dataset are shown in the form of a confusion matrix in Fig. 4b. Detailed quantitative results can be found in Tabs. 2 and 4.

3.3 Evaluation Matrices

There are several methods for evaluating the performance of the classification model, but the most common metrics are precision, recall, and accuracy. These are represented mathematically in Eqs. (4)–(6).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where true positive (TP) is the number of violent activities correctly identified; false positive (FP) is the number of non-violent activities incorrectly predicted as violent; true negative (TN) is the number of non-violent activities correctly identified; and false negative (FN) is the number of violent activities that are incorrectly predicted as non-violent.

Figure 3: Example frames from the hockey fight and violent crowd datasets. Images shown in the first section are scenes from the hockey fight dataset, while those in the second section are scenes from the violent crowd dataset

Table 1: Description and statistics for the datasets

Dataset	Samples	Resolution	Violent scenes		Non-violent scenes	
			No. of clips	Frame rate	No. of clips	Frame rate
Hockey fight [44]	1000	360 × 288	500	25	500	25
Violent crowd [8]	246	320 × 240	123	25	123	25

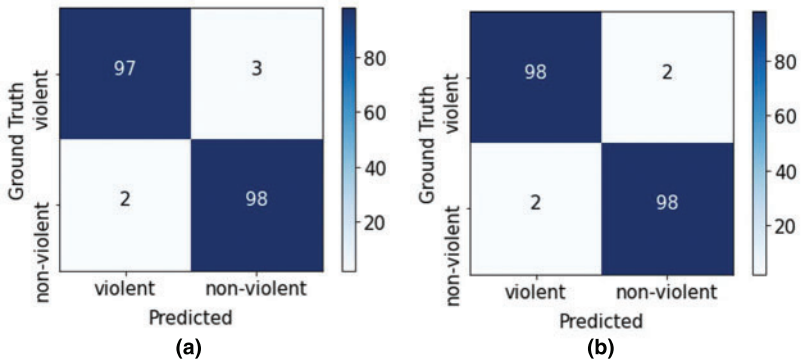

Figure 4: Confusion matrices for (a) The hockey fight dataset, and (b) The violent crowd dataset

Table 2: Comparison of the proposed method with other state-of-the-art methods on the Violent Crowd and Hockey Fight datasets

Methods	Accuracy (%) of method on each dataset	
	Violent crowd	Hockey fight
ViF, OViF, AdaBoost and SVM [11]	88	87.50
Hough forests and 2D CNN [29]	–	94.6
ViF [8]	81.3	82.90
Improved fisher vectors [18]	96.4	93.7
3DCNN [7]	98	96
Proposed method	99.89	98.80

3.4 Results and Discussion

In a CNN, the most challenging tasks involve finding the optimal kernel size, the optimal number of filters per layer, and the optimal formation of the layers. These hyperparameters are highly correlated with the input data. The hidden layers in CNN architecture act as a black box, meaning that it is very difficult to identify the correct number of filters and formation of layers directly, and we therefore used a heuristic approach to develop an efficient model. We performed multiple experiments with different numbers of convolutional layers, hyperparameter settings, activation functions, learning rates and temporal information. To efficiently learn the patterns of violent and non-violent activity, we need to input massive amounts of labelled data to train the deep architecture. Each violent activity dataset contains a number N of short clips with different durations, and each video in the dataset belongs to one of two categories: violent

or non-violent. During training, each individual input video clip is fed to the 3DCNN as a set of batches of length 16 frames, to allow the model to learn the temporal features of the input video.

3.4.1 Analyses of Different Learning Rates

When training a neural network, the most important hyperparameter is the learning rate, and the choice of the optimal learning rate greatly affects the generalisation of the deep learning model. There are two key assumptions that should be taken into consideration when selecting the learning rate during training. Firstly, the learning rate should not be too high, as this will cause overshoot while finding the minimal points and the model will allocate large updates to the weights, causing divergent behaviour. At the beginning of our experiments, we therefore used a low learning rate of 0.00001, as shown in Fig. 6a. However, from our results, we found that a low learning rate did not always perform very well. We therefore performed further experiments based on the second key assumption, in which the learning rate should not be too low since numerous updates will be required before the optimal point is reached. We finally applied a learning rate of 0.001 with the Adam optimiser, and achieved state-of-the-art results. The highest values of accuracy were 98% for the Hockey Fight dataset and 99% for the Violent Crowd dataset, as shown in Fig. 6b. It can be seen that the highest accuracy of 98% was obtained with a learning rate of 0.001 over 500 epochs. After conducting numerous experiments, we observed that changing the learning rate affected the loss, accuracy, and number of iterations. Fig. 6a shows that the changes in loss and accuracy are strongly related to the variation in the learning rate. For instance, for the first 100 epochs, the training loss, validation loss, training accuracy and validation accuracy were 0.299, 0.218, 88.3% and 94.5%, respectively. Over time, as the number of iterations increased, the model loss decreased while the accuracy increased. Finally, at 500 epochs, the loss was reduced to 0.02 and the accuracy reached 99.44% when the learning rate was set to 0.00001 and the experimental setup, training loss, validation loss, training accuracy and validation were kept at the same values as before. The accuracy for the first hundred epochs was 94%, while at 500 epochs, the obtained training loss, validation loss, training accuracy, and validation accuracy were 0.01, 0.15, 99%, and 98%, as shown in Fig. 6b.

We also trained our model with the same setup on the Violent Crowd dataset, and the results of the training and validation process are shown in Fig. 6a. The pink line indicates the experiment in which we achieved the highest accuracy of 99% and a loss of 0.22 over 500 epochs.

3.4.2 Computational Complexity Analysis

A CCN architecture consists of three different types of layers: convolutional, max-pooling and fully connected layers. The convolutional layers share a filter of fixed size, which is used to extract spatial features. The max-pooling layer does not learn anything during training. However, the fully connected layer is extremely expensive in terms of computation, as it involves one-to-one connectivity for each pixel in an image. When a depth channel or a temporal dimension is added, a 3DCNN becomes computationally more complex than a classical 2DCNN architecture. For instance, in [7], the authors presented a 3DCNN architecture for recognition of violent activity in a surveillance system. At the feature extraction stage, eight convolutional layers were used, feature reduction was achieved through five max-pooling layers, and the classification stage used two fully connected layers with 4096 neurons in each. To limit the computational complexity of our model, we conducted multiple experiments and finally developed a lightweight 3DCNN architecture consisting of eight convolutional layers, three max-pooling layers, and one fully connected layer with 512 neurons rather than 4,096. The proposed network involved 14,352,811 trainable parameters,

whereas the network in [7] involved 22,228,802, a reduction of 7,875,991. The final result was a computationally inexpensive model that was capable of achieving high accuracy compared to state-of-the-art models, as shown in Tab. 2. We reduced the overall complexity of our proposed network by using only one fully connected layer.

3.4.3 Ablation Study of Different Sequences

Activity recognition with a 2DCNN involves extracting only the spatial features from video frames, which is insufficient for activity recognition. Khan et al. [45] proposed a CNN architecture called MobileNet for violence detection from movies that only extracted spatial features from videos, whereas a 3DCNN extracts both spatial and temporal features from videos. Temporal or sequence information plays a critical role in activity recognition. In view of this, we conducted several experiments to analyse temporal information, as shown in Fig. 5. When the number of sequences was increased, the model efficiently learned the data, but there was a trade-off between the number of sequences and the computation time. To find the optimal length of a sequence, we tested our method with different sequence lengths ranging from six to 16 consecutive frames. The experimental results are shown in Fig. 5, and it can be observed that the lowest accuracy was achieved for a sequence of six frames. As the length of the sequence was increased, the accuracy improved. The highest values of accuracy for both the Hockey Fight (98%) and Violent Crowd datasets (99%) were achieved for a sequence of 16 frames. We evaluated our proposed system using the metrics of precision, recall, and accuracy, and the results are listed in Tab. 2. For the Hockey Fight dataset, the highest accuracy was 98% with an F1-score of 0.97, while for the Violent Crowd dataset, the highest accuracy was 99% with an F1-score of 0.98.

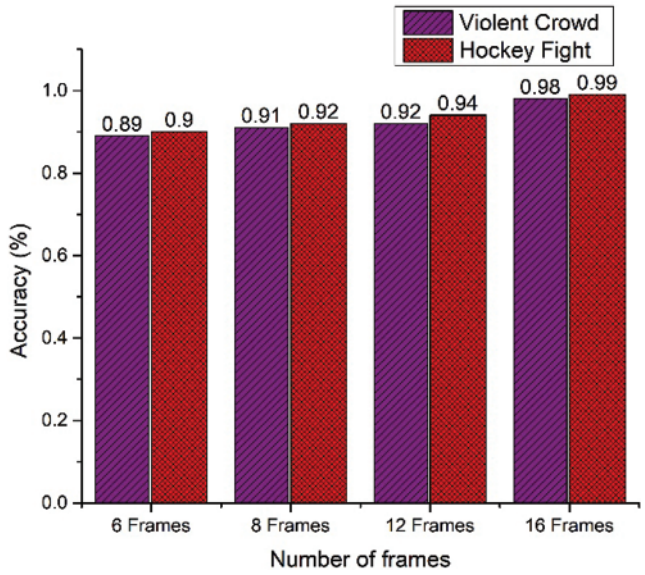

Figure 5: Temporal analysis of the proposed 3DCNN architecture in terms of accuracy on the hockey fight and violent crowd datasets

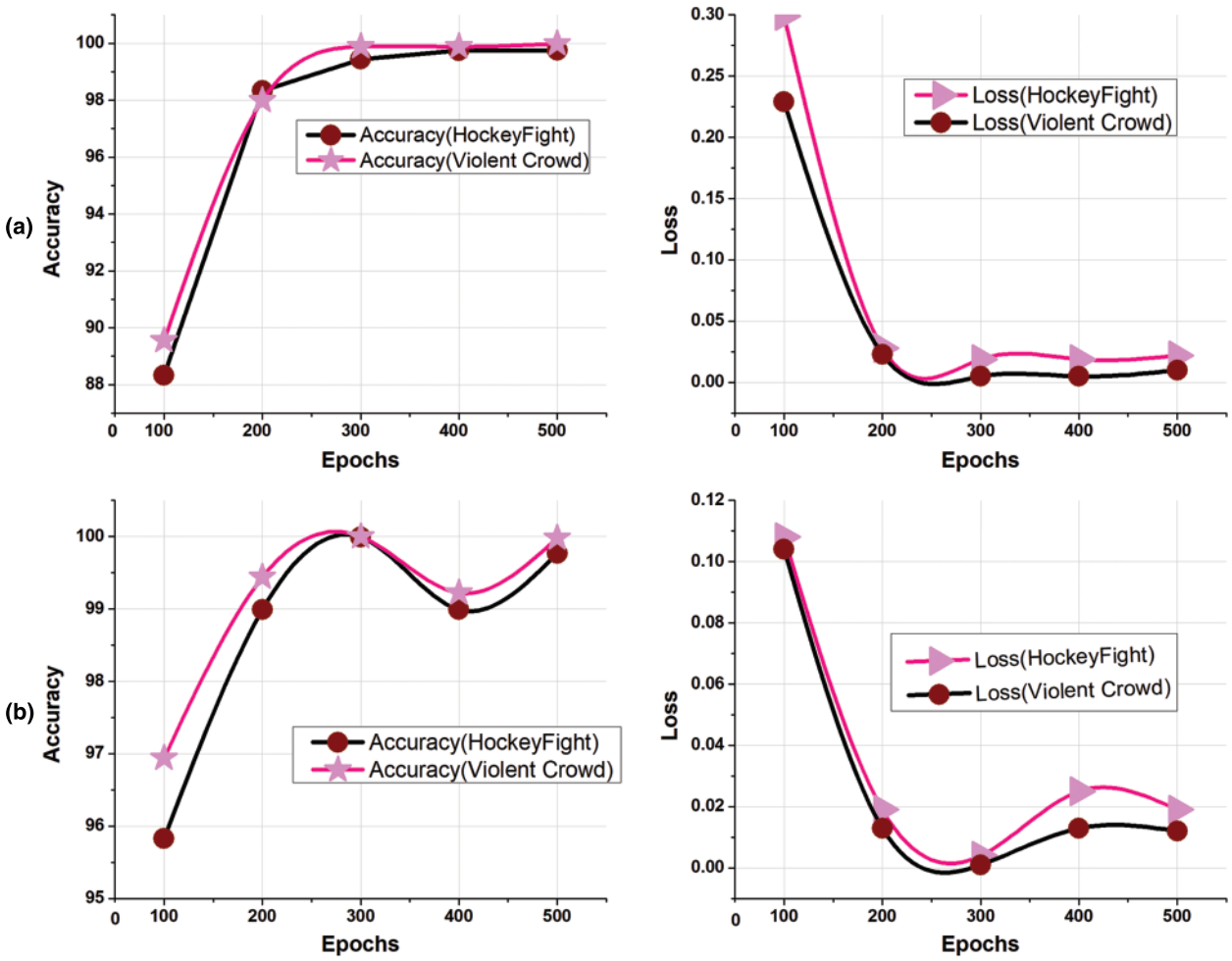

Figure 6: Impact of different learning rates on the hockey fight and violent crowd datasets: (a) the optimal learning rate of 0.001 with the Adam optimiser, and (b) a smaller learning rate of 0.00001 with the Adam optimiser

3.5 Comparative Analysis

In this section, we compare the results obtained from our proposed 3DCNN architecture with those of state-of-art methods. A comparative analysis is shown in Tab. 2. The first row shows the results obtained by the method proposed in [11], which used violent flows descriptor to estimate the magnitude of the violence and the AdaBoost algorithm for feature extraction. Finally, an SVM was deployed as a violent activity classifier.

Their method achieved an accuracy of 87.50% on the Hockey Fight dataset and 88% for the Violent Crowd dataset. Recently, Serrano et al. [29] used Hough forests with a 2DCNN to detect violent activity, and this approach gave an accuracy of 94.6% on the Hockey Fight dataset. Similarly, Hassner et al. [8] experimented on the same datasets with a scheme that used ViF for feature extraction and SVM for classification, and obtained accuracies of 82.90% and 81.3%, respectively. A sliding window method was used with an improved Fisher vector in [18], and this

approach achieved values of 93.7% and 96.4% for accuracy, respectively. In [7], the author used a 3DCNN and obtained accuracy scores of 96% and 98% on the Hockey Fight and Violent Crowd datasets, respectively. The last row of Tab. 2 shows the performance of the proposed model. In general, the computational complexity of CNN architectures is directly related to the fully connected layer. For instance, the method presented in [7] used eight convolutional layers and two fully connected layers with 4096 neurons per layer. In contrast, our proposed 3DCNN architecture consists of eight convolution layers and only one fully connected layer with 512 neurons. Our model therefore performed favourably against the scheme [7] in terms of the accuracy and network complexity.

3.6 Quantitative Evaluation of Our Proposed Camera Prioritisation Method with and without a Weighting Strategy

For camera prioritisation, there is no standard evaluation matrix to check their performance. However, to achieve the best performance, the camera prioritisation evaluation metrics should maximise the impact of the overall system accuracy and consider the vulnerable position of the camera node. This is very important, since some locations such as banks and border control stations require a high level of surveillance, based on selecting camera nodes with a high priority. Video streams from camera nodes are fed to a trained 3DCNN model to classify and calculate the probability of violent or non-violent activity. If violent activity is detected, the probability calculated by the trained 3DCNN model and the metadata about the camera are passed to the LTGS module, as shown in Eqs. (2) and (3). The weights or metadata reflect the importance of the location of the node: a low weight means that the position is not important, while a higher weight indicates that the position of the node is more important. A mathematical representation of our camera prioritisation scheme is given in Eqs. (2) and (3). Tab. 3 presents the results of our experiments without a weighting strategy, in which the weight is set to a constant value of one. In Tab. 3, the column marked 'Ground truth' represents the number of cameras showing violent activity, which need to be prioritised. In contrast, the column marked 'Cameras prioritised by the model' shows the number of cameras selected for prioritisation by our proposed system. In the first experiment, we used 10 cameras over which violent activity was randomly distributed to check the robustness of our proposed model on a small network of WVSNs. In the first test, cameras 1 to 8 captured violent scenes, while cameras 9 and 10 recorded normal events. When we passed these streams to the trained 3DCNN model, it prioritised seven of the eight cameras that recorded these activities. To calculate the overall accuracy, we divided the total correct number of prioritised cameras by the total number of cameras that should have been prioritised. The proposed model achieved an average accuracy of 86.1%.

Table 3: Experiments on camera prioritisation with a constant weight value (w = 1)

Experiments	Number of cameras	Ground truth	Cameras prioritised by the model (φ)	Accuracy (%)
Experiment 1				
Test 1	10	8	7	87.5
Test 2	10	7	6	85.7
Test 3	10	9	7	77.7
Test 4	10	6	6	1
Test 5	10	5	4	80
Average	10	7	6	86.1

(Continued)

Table 3: Continued

Experiments	Number of cameras	Ground truth	Cameras prioritised by the model (φ)	Accuracy (%)
Experiment 2				
Test 1	15	13	11	84.6
Test 2	15	11	10	90.9
Test 3	15	9	9	1
Test 4	15	14	14	1
Test 5	15	8	6	75
Average	15	11	10	90.1
Experiment 3				
Test 1	20	14	14	1
Test 2	20	15	14	93.3
Test 3	20	16	15	93.7
Test 4	20	13	13	1
Test 5	20	17	14	82.3
Average	20	15	14	93.8

In the second experiment, the number of cameras was increased from 10 to 15, and the total number of cameras that needed to be prioritised was 11. For each individual test in this set, the number of cameras reporting violent activity is listed under the 'Ground truth' column. In this experiment, our proposed system prioritised 10 cameras, giving an average accuracy of 90.1%. Similarly, in experiment 3, we evaluated 20 cameras and achieved an accuracy of 93.8%. In experiment 1, our model prioritised all 14 cameras and we achieved an accuracy of 100%. The details of experiments 2–5 are shown in Tab. 3. The results of the experiments with a weighting strategy are presented in Tab. 4.

3.7 Result Derivatives

Existing systems use highly intensive computational methods for the recognition of abnormal activity, as discussed in Sections 1.1 and 1.2, and cannot be deployed in real-time surveillance systems. Techniques in the literature on camera prioritisation are mostly based on handcrafted features, and no statistical analyses of their results are provided. Furthermore, no discussion of the validity of camera prioritisation is available for large-scale WVSNs. In this paper, we have proposed a lightweight 3DCNN model for abnormal activity recognition using resource-constrained devices, which outperformed existing state-of-art methods in terms of accuracy and F1-score. Furthermore, we have proposed an intelligent algorithm called LTGS that prioritises cameras showing abnormal activity above those showing a normal stream. The results of multiple comprehensive experiments and evaluations show that the proposed system performs well in terms of both violent activity recognition and efficient camera prioritisation in large-scale surveillance networks.

Table 4: Experiments on camera prioritisation with a weighting strategy

Experiments	Number of cameras	Ground truth	Cameras prioritised by the model	Weight values (w)	Accuracy (%)	Total net accuracy (φ)
Experiment 1						
Test 1	10	8	7	4	3.5	97
Test 2	10	7	6	5	4.625	99
Test 3	10	9	7	3	2.571	92.8
Test 4	10	6	6	1	1	73.1
Test 5	10	5	4	2	1.6	83.2
Average	10	7	6	3	2.659	89.02
Experiment 2						
Test 1	15	13	11	2	1.692	84.4
Test 2	15	11	10	3	2.727	93.8
Test 3	15	9	9	1	1	73.1
Test 4	15	14	14	2	2	88
Test 5	15	8	6	2	1.5	81.7
Average	15	11	10	2	1.7838	84.2
Experiment 3						
Test 1	20	14	14	1	1	73.1
Test 2	20	15	14	4	3.732	97.6
Test 3	20	16	15	6	5.622	99.6
Test 4	20	13	13	2	2	88
Test 5	20	17	14	2	1.646	83.8
Average	20	15	14	3	2.8	88.42

4 Concluding Remarks and Directions for Future Work

In this study, we have investigated the strength and capabilities of a 3DCNN for intelligent camera prioritisation in large-scale WVSNs, based on violent activity recognition. WVSNs typically consist of a large number of visual nodes, each of which continuously generates a massive amount of video data. The efficient monitoring and streaming of such huge amounts of data is very challenging, due to the limited availability of computational resources. To overcome the drawbacks of traditional surveillance systems, we have proposed a novel intelligent camera prioritisation framework for large-scale WVSNs. A 3DCNN is used for violent activity recognition that is capable of learning not only spatial but also temporal information. The proposed model can prioritise cameras with an accuracy of 98%–99%. In future work, we aim to further reduce the cost of the proposed framework by customising the Raspberry Pi, for example by removing unwanted hardware like the mouse, keyboard and GPIO pens. Removing unnecessary functionalities from this device can significantly reduce the overall computational complexity. For activity recognition, the features extracted by our proposed architecture could be fed to a recurrent neural network such as LSTM to classify the input video stream more efficiently.

Acknowledgement: This work was supported by the faculty research fund of Sejong University in 2020 and also supported by Institute of Information & communications Technology Planning &

Evaluation (IITP) grant funded by the Korea government (MSIT) (2019-0-00136, Development of AI-Convergence Technologies for Smart City Industry Productivity Innovation).

Funding Statement: Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2019-0-00136, Development of AI-Convergence Technologies for Smart City Industry Productivity Innovation).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad and S. W. Baik, "CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks," *Multimedia Tools and Applications*, vol. 1, pp. 1–17, 2020.

[2] K. Karboub, M. Tabaa, S. Dellagi, A. Dandache and F. Moutaouakkil, "Intelligent patient monitoring for arrhythmia and congestive failure patients using internet of things and convolutional neural network," in *31st Int. Conf. on Microelectronics*, Cairo, Egypt, pp. 292–295, 2019.

[3] Y.-H. Tsai, J.-K. Hsu, Y.-E. Wu and W.-F. Huang, "Distributed multimedia content processing in ONVIF surveillance system," in *2011 Int. Conf. on Future Computer Sciences and Application*, Hong Kong, China, pp. 70–73, 2011.

[4] C. A. T. Stelios, D. Stelios and D. Antonios, "Automated real-time risk assessment for airport passengers using a deep learning architecture," in *Signal Processing, Sensor/Information Fusion and Target Recognition XXVIII*, Maryland, United States, pp. 110180, 2019.

[5] D. Arjun, P. K. Indukala and K. A. U. Menon, "PANCHENDRIYA: A Multi-sensing framework through wireless sensor networks for advanced border surveillance and human intruder detection," in *2019 Int. Conf. on Communication and Electronics Systems*, Coimbatore, India, pp. 295–298, 2019.

[6] S. Telang, A. Chel, A. Nemade and G. Kaushik, "Intelligent Transport System for a Smart City," in *Security and Privacy Applications for Smart City Development*. Cham: Springer, pp. 171–187, 2021.

[7] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq and S. W. Baik, "Violence detection using spatiotemporal features with 3D convolutional neural network," *Sensors*, vol. 19, no. 11, pp. 2472, 2019.

[8] T. Hassner, Y. Itcher and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, Providence, RI, USA, pp. 1–6, 2012.

[9] J.-F. Huang and S.-L. Chen, "Detection of violent crowd behavior based on statistical characteristics of the optical flow," in *11th Int. Conf. on Fuzzy Systems and Knowledge Discovery*, Xiamen, China, pp. 565–569, 2014.

[10] T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang *et al.,* "A new method for violence detection in surveillance scenes," *Multimedia Tools and Applications*, vol. 75, no. 12, pp. 7327–7349, 2016.

[11] Y. Gao, H. Liu, X. Sun, C. Wang and Y. Liu, "Violence detection using oriented VIolent flows," *Image and Vision Computing*, vol. 48–49, no. 6, pp. 37–41, 2016.

[12] D. Chen, H. Wactlar, M.-Y. Chen, C. Gao, A. Bharucha *et al.,* "Recognition of aggressive human behavior using binary local motion descriptors," in *30th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, Vancouver, BC, Canada, pp. 5238–5241, 2008.

[13] F. D. De Souza, G. C. Chavez, E. A. do Valle Jr and A. D. A. Araújo, "Violence detection in video using spatio-temporal features," in *23rd SIBGRAPI Conf. on Graphics, Patterns and Images*, Gramado, Brazil, pp. 224–230, 2010.

[14] L. Xu, C. Gong, J. Yang, Q. Wu and L. Yao, "Violent video detection based on MoSIFT feature and sparse coding," in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, pp. 3538–3542, 2014.

[15] K. Lloyd, P. L. Rosin, D. Marshall and S. C. Moore, "Detecting violent and abnormal crowd activity using temporal analysis of grey level co-occurrence matrix (GLCM)-based texture measures," *Machine Vision and Applications*, vol. 28, no. 3–4, pp. 361–371, 2017.

[16] E. Y. Fu, H. V. Leong, G. Ngai and S. C. Chan, "Automatic fight detection in surveillance videos," *International Journal of Pervasive Computing and Communications*, vol. 3, pp. 1–11, 2017.

[17] J. Mahmoodi and A. Salajeghe, "A classification method based on optical flow for violence detection," *Expert Systems with Applications*, vol. 127, no. 1, pp. 121–127, 2019.

[18] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *2016 13th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Colorado Springs, CO, USA, pp. 30–36, 2016.

[19] A. Deshmukh, K. Warang, Y. Pente and N. Marathe, "Violence detection through surveillance system," in *ICT Systems and Sustainability*. Berlin, Germany: Springer, pp. 503–511, 2021.

[20] R. Maqsood, U. I. Bajwa, G. Saleem, R. H. Raza *et al.,* "Anomaly Recognition from surveillance videos using 3D convolutional neural networks," *Multimedia Tools and Applications*, 2017. https://doi.org/10.1007/s11042-021-10570-3.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Montreal, Canada, pp. 4489–4497, 2015.

[22] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Nashville, United States, pp. 6299–6308, 2017.

[23] A. Diba, M. Fayyaz, V. Sharma, A. Hossein Karami, M. Mahdi Arzani *et al.,* "Temporal 3d convnets using temporal transition layer," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Nashville, United States, pp. 1117–1121, 2018.

[24] Z. Qiu, T. Yao and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Montreal, Canada, pp. 5533–5541, 2017.

[25] D. Tran, J. Ray, Z. Shou, S.-F. Chang and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *ArXiv*, vol. abs/1708.05038, 2017. https://arxiv.org/abs/1708.05038.

[26] J. Stroud, D. Ross, C. Sun, J. Deng and R. Sukthankar, "D3d: Distilled 3d networks for video action recognition," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, Waikoloa, United States, pp. 625–634, 2020.

[27] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun *et al.,* "A closer look at spatiotemporal convolutions for action recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Nashville, United States, pp. 6450–6459, 2018.

[28] S. Xie, C. Sun, J. Huang, Z. Tu and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. of the European Conf. on Computer Vision*, Montreal, Canada, pp. 305–321, 2018.

[29] I. Serrano, O. Deniz, J. L. Espinosa-Aranda and G. Bueno, "Fight recognition in video using hough forests and 2D convolutional neural network," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4787–4797, 2018.

[30] A. Ullah, K. Muhammad, K. Haydarov, I. U. Haq, M. Lee *et al.,* "One-shot learning for surveillance anomaly recognition using siamese 3D CNN," in *2020 Int. Joint Conf. on Neural Networks*, Glasgow, UK, pp. 1–8, 2020.

[31] S. Maresca, G. Serafino, F. Scotti, F. Amato, L. Lembo *et al.,* "Photonics for coherent MIMO radar: An experimental multi-target surveillance scenario," in *2019 20th Int. Radar Symp.*, Ulm, Germany, pp. 1–6, 2019.

[32] N. Kumar and N. Sukavanam, "A cascaded CNN model for multiple human tracking and re-localization in complex video sequences with large displacement," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6109–6134, 2020.

[33] C. Sampedro, A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente *et al.,* "A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 601–627, 2019.

[34] S. S. Sengar and S. Mukhopadhyay, "Moving object area detection using normalized self adaptive optical flow," *Optik*, vol. 127, no. 16, pp. 6258–6267, 2016.

[35] J. Dou, Q. Qin and Z. Tu, "Background subtraction based on circulant matrix," *Signal, Image and Video Processing*, vol. 11, no. 3, pp. 407–414, 2017.

[36] M. Fei, J. Li and H. Liu, "Visual tracking based on improved foreground detection and perceptual hashing," *Neurocomputing*, vol. 152, no. 8, pp. 413–428, 2015.

[37] A. Zam, M. R. Khayyambashi and A. Bohlooli, "Energy-aware strategy for collaborative target-detection in wireless multimedia sensor network," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18921–18941, 2019.

[38] I. Mehmood, M. Sajjad, W. Ejaz and S. W. Baik, "Saliency-directed prioritization of visual data in wireless surveillance networks," *Information Fusion*, vol. 24, no. 3, pp. 16–30, 2015.

[39] S. S. Thomas, S. Gupta and V. K. Subramanian, "Event detection on roads using perceptual video summarization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2944–2954, 2018.

[40] K. Muhammad, T. Hussain and S. W. Baik, "Efficient CNN based summarization of surveillance videos for resource-constrained devices," *Pattern Recognition Letters*, vol. 130, pp. 370–375, 2020.

[41] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy *et al.,* "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, no. 11, pp. 354–377, 2018.

[42] A. Piergiovanni, A. Angelova and M. S. Ryoo, "Evolving losses for unsupervised video representation learning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Shanghai, China, pp. 133–142, 2020.

[43] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, S. Long *et al.,* "Caffe: Convolutional architecture for fast feature embedding," in *Proc. of the 22nd ACM Int. Conf. on Multimedia*, Florida, USA, pp. 675–678, 2014.

[44] E. B. Nievas, O. D. Suarez, G. B. García and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Int. Conf. on Computer Analysis of Images and Patterns*, NY, USA, pp. 332–339, 2011.

[45] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik and M. Y. Lee, "Cover the violence: A novel deep-learning-based approach towards violence-detection in movies," *Applied Sciences*, vol. 9, no. 22, pp. 4963, 2019.