An aggregated deep convolutional recurrent model for event based surveillance video summarisation: A supervised approach

Sreeja M. U. | Binsu C. Kovoor

Division of Information Technology, Cochin University of Science and Technology, Kochi, Kerala, India

Correspondence
Sreeja M. U., Division of Information Technology, Cochin University of Science and Technology, Kochi, Kerala, India.
Email: kishan.sreeja@gmail.com

Abstract
Surveillance video summarisation is characterised by extracting video segments containing abnormal events from surveillance video footages. Accurate identification of abnormal events from surveillance footages is of paramount importance in surveillance video summarisation. Accordingly, the proposed framework builds an aggregated convolutional recurrent model that can precisely detect the suspicious events in a surveillance footage, by employing a supervised learning which is found to yield better results compared with unsupervised counterparts. The preliminary stage in the model is a multilayer Convolutional Neural Network for frame-level feature extraction followed by stacked bidirectional Gated Recurrent Unit for sequence-level feature extraction and classification. Since the video clips used for training are not implicit to surveillance, a block-based approach for testing on surveillance videos is proposed. The results evaluated on two custom datasets, Streets and Campus, prove that the proposed model produces remarkable results leveraging the properties of bidirectional GRUs with supervised learning. Extensive experimental analysis on selection of optimum architecture is conducted which substantiates the significance of stacked bidirectional GRUs over unidirectional ones. Additionally, qualitative results ensure that summaries produced are concise, representative, complete, diverse and informative. Moreover, comparison of the performance of the proposed model with state of the art certainly proves the superiority of the proposed model.

1 | INTRODUCTION

Video surveillance systems are the most promising form of security technology as they can be used to monitor areas from remote locations using recording cameras connected to an IP network. The prevalence of surveillance systems stems from the fact that they are economical and at the same time has been proven to be beneficial for preventing and resolving crimes effectively. However, surveillance cameras record day long footages resulting in the generation of huge amount of videos. According to recent reports, the amount of data produced by surveillance cameras worldwide per day is nearly 700 petabytes and is expected to increase exponentially in the coming years. Apparently, browsing and finding relevant information from these long footages has become a tedious task. Automatic identification and extraction of the exact relevant information require sophisticated technology for processing these videos. Video summarisation aids here by automatically identifying the video segments with unusual events such as crimes, a security breach, vandalism, etc. An ideal summary should contain the maximum significant information from the surveillance video in the shortest possible representation.

Recent research has emphasised the significance of employing summarisation models in a category-specific manner based on the properties of a video genre to yield better summaries [1, 2]. Unlike other genre of videos, the primary characteristics of surveillance videos are that the cameras remain stationary and that most of the time there are no significant events present in the videos. Hence, exactly identifying the segments of the surveillance videos where an unusual event has occurred is of utmost importance. Several models for event detection based on optical flow, motion and object detection have been addressed in the literature and have shown successful results. Regardless, deep learning has largely
influenced all domains of computer vision lately, not to mention video summarisation. In fact, the effectiveness of deep learning in surveillance video summarisation can be assessed by the amazing results achieved in the recent works.

Although, unsupervised approaches are gaining high popularity, the results obtained by the supervised methods are still superior. The summaries produced are often static or dynamic in nature. Dynamic summaries are defined as the combination of identified key sequences from the video whereas static summaries are formed by combining stationary key frames. Considering the significance of events in surveillance videos and from the recent research, it is evident that the dynamic summaries are ideal for surveillance videos, since an event can far better be represented by a video sequence than its key-frame counterpart.

Surveillance video summarisation techniques employing supervised event detection algorithms have not been addressed in the recent literature to the best of our knowledge. To this end, the proposed framework builds an aggregated deep convolutional recurrent model for surveillance video summarisation based on unusual event detection. Since temporal information in videos is paramount in determining if an event is unusual or not, specialised stacked bidirectional gated recurrent units (GRU) are employed for the video sequence processing, whereas the frame level details are extracted by a multi-layer convolutional neural network. Bidirectional RNNs do not only consider the temporal dependencies in forward direction, but also in the backward direction which accounts to the superior performance of the model. The performance improvement further obtained by using stacked layers of bidirectional GRUs owes to the fact that deeper the network, greater the performance to a certain degree. Additionally, a block-based strategy is put forth for testing on real surveillance videos which additionally enhance the summarisation performance. The major highlights of the proposed framework are:

1. An aggregated model combining a multi-layer CNN and stacked bidirectional GRUs for surveillance video summarisation based on supervised event classification.
2. A unique model capable of identifying the predominant abnormal events present in the surveillance videos from the minimal number of classes trained.
3. An improved block-based pre-processing strategy for relevance computation and summary generation from actual surveillance videos.
4. A generic custom video clip data augmentation and generation approach for optimal training performance.
5. Two surveillance video datasets for experimentation and performance evaluation in the surveillance video genre.

The rest of the paper is organised as follows. Section 2 reviews the state-of-the-art models in the area of surveillance video summarisation. Section 3 outlines the challenges and motivation that led to the design of the proposed framework. The detailed methodology along with the network architecture are described in section 4. Experiments conducted and the results obtained are analysed in section 5 followed by concluding remarks and future directions in section 6.

2 | LITERATURE REVIEW

This section begins by reviewing the elementary techniques applied for surveillance video summarisation apart from deep learning. This is followed by how deep learning has impacted surveillance video summarisation in particular and subsequently, how video summarisation benefits from anomaly detection techniques specifically in the surveillance domain.

2.1 | Surveillance video summarisation frameworks

Summarisation frameworks using object motion analysis for surveillance videos is discussed in this section. Thomas et al. [3] designs a novel stage in the surveillance video summarisation which is an accident stage study with the type of collision. Accident detection is performed from a stack of videos using optimization problem and submodularity. Yet another model for abnormal event detection based on an improved optical flow method, specifically for traffic video summarisation is developed by Athanasious et al. [4]. The method incorporated is super-oriented optical flow clustering. Thomas et al. [5] considers human attention for surveillance video summarisation. Human attention model is used for detecting key events. HVS (Human Visual System) is the system used for finding salient regions in the frames.

Multi view videos are the videos of the same scene that are captured by different cameras at different views. A unique video summarisation framework for multi view videos using DNA sequence simulation is proposed by Kumar and Shrimankar [6]. Four phases are: visual feature extraction, nucleotide sequence formation, multi view video correlation with FASTA and event summarisation. Sobhani et al. [7] develops an ontology for formally representing the major events in a surveillance system which is extended to forensics too. The ontology is built on the famous DOLCE ontology which is extensively used for linguistic and cognitive modelling of knowledge. Kahar and Izquierdo [8] also builds an ontology that aids in identifying major events in CCTV videos. The events for surveillance videos such as crime, attack, smashing, etc. are inferred from a set of rules designed for the ontology. Nasir et al. [9] proposes video summarisation model that can cost effectively be deployed on local networks with the help of fog computing.

2.2 | Deep learning-based video summarisation

Deep learning-based models for video summarisation with emphasis on surveillance genre are discussed in this section. An unsupervised video summarisation based on a generative
adversarial framework is implemented by Mahasseni et al. [10]. The model consists of a summariser and a discriminator where the summariser is an LSTM autoencoder whose aim is to learn the latent input representation and the discriminator is again an LSTM classifier used to distinguish the summarised outputs from real inputs. Panagiotakis et al. [11] implemented a summarisation model based explicitly on user preferences. The model extracts personalised video segments without the conventional audio-visual processing. The model makes use of a segment duration factor clubbed with a Synthetic coordinate based recommender system for detecting the key segments. Roohan et al. [12] employs a fully convolutional neural network which does not utilise an RNN to label video sequences. The functions constituted in the model are convolution, pooling and deconvolution across time. Zhao et al. [13] proposes a hierarchical recurrent model that can be adapted to the video structure irrespective of the length of each shots. The first bidirectional LSTMs slides over video frames and detects shot boundaries and the second one computes the summary probability. The hierarchical structure ensures that the structural information of videos is preserved. Muhammad et al. [14] utilises convolutional neural network for efficient surveillance video summarisation. Within each shot, the frame with the highest memory and entropy score is chosen as the keyframe. Zhang et al. [15] utilises online motion autoencoder for object level summarisation. The autoencoder is a stacked sparse LSTM network. A novel surveillance video dataset named Orangeville is introduced. Deep CNN with hierarchical weighted fusion for summarisation of surveillance videos captured in IoT settings is put forward by Muhammad et al. [16]. Muhammad et al. [17] extends the model for industrial surveillance by implementing a deep learning-based system for surveillance video summarisation using Industrial IoT. The video is captured using resource constraint devices where a coarse refinement is done and sequence-level feature extraction is done in the latter stage using the deep learning model. Key frame selection based on the dissimilarity measure computed by f-divergence asymmetric strategy for multiple change point detection is proposed by Gao et al. [18] which is used to segment a video into non-overlapping clips. Liu et al. [19] proposes an LSTM based event detection in surveillance videos. Experiments are conducted for recognising multiple events in a given surveillance video by sequence to sequence network without object detection. Lu et al. [20] develops an unsupervised approach for keyframe selection for efficient and scalable summarisation of surveillance videos. The model requires no training or learning. It provides an analysis of the video content by measuring abnormalities in a video using the temporal video curve and typical regression model. Lei et al. [21] develops an action based video summarisation model based on reinforcement learning. Weakly annotated model is used for action parsing in videos and later a deep RNN selects key frames by multiple instance learning. Generative Adversarial networks along with a novel anomaly detection framework is described by Ammar et al. [22] for segmenting and classifying the moving objects in surveillance video sequences.

### 2.3 Video summarisation based on anomaly detection

Recently, a number of works have addressed the importance of anomaly detection in video summarisation, especially in the surveillance domain. Ratre and Pankajakshan [23] have recently developed an anomalous object detection and localisation model using the Tucker tensor decomposition. For object detections, the model utilises the background subtraction algorithm, Gaussian mixture model and later on, the object classification is performed based on the Tucker tensor decomposition and object localisation based on the shape and speed which result in anomaly detection. Sultani et al. [24] focuses on detecting anomalies for processing real-world surveillance videos. The model proposed is a weakly supervised one in the sense that the labelled data for anomaly detection are at video level rather than clip level. Three dimensional convolutional networks are used for feature extraction from multiple instances. Luo [25] describes a deep learning based anomaly detection model that uses the sparse coding technique which acts similar to specialised recurrent neural networks. The specialised networks are used in conjunction with autoencoders to infer the relationship between the neighbouring frames in a normal event as well as in an anomalous event. Li and Chang [26] implements a generative adversarial network for anomaly detection in surveillance videos. The idea is that the latent representation for the generator is based on the fact that they differ for normal and anomalous videos. The novel deep learning-based model also considers two types of anomaly scores, one based on the motion and the other based on the appearance. Nawaratne et al. [27] puts forth a real time anomaly detection approach based on the concept of deep and active learning. An incremental spatiotemporal learner in an unsupervised manner is employed for detecting anomalies by learning the kind of normality. The classification model employed is fuzzy and anomaly detection is performed in real time with the help of active learning. Doshi and Yilmaz [28] proposed a fast anomaly detection method which has the least computational time. The model employs deep learning for the object detection alone and the decision-making process for anomalous segment detection is based purely on machine learning algorithms; nearest neighbour and k-means algorithm.

### 3 MOTIVATION

In light of the recent literature, the following facts are elicited:

1. Although, a large number of summarisation models explicitly for surveillance video domain have been addressed in the literature, supervised models that has been proven to produce superior results is not much-explored. Given the recent progress, it is high time a deep learning model that leverages the benefit of supervised learning for abnormal event identification for surveillance video summarisation has been addressed.
2. Efforts to build models to detect the event classes identified in the literature as abnormal or suspicious events such as hit, run, gun shoot, kicking, punching, etc. [7, 8], are minimal. The proposed model is an attempt in this direction to train a model capable of identifying the most common abnormal events in a surveillance scenario.

3. Various datasets have been put forth in the literature for experimenting in the surveillance video summarisation domain. However, it can be noted that neither the common events occurring generically in streets and campuses which contribute the most towards surveillance video category have been paid much attention nor datasets for experimenting in this category was available in the literature. The need for public datasets in streets and campus domain manifests from the detailed analysis of the publicly available datasets.

4. Besides, computational overhead is a prime concern in the surveillance video summarisation spectrum. Attempts to employ a supervised model with minimal number of event classes, that is also capable of identifying a wide range of other abnormal events as well in the surveillance domain has not been addressed in the literature. This led to the identification of the aforementioned classes of abnormal events for surveillance video summarisation.

4 | THE PROPOSED METHODOLOGY

The proposed framework formulates surveillance video summarisation as a sequence classification problem where the input is a surveillance video sequence and the output determines whether the sequence is relevant to the video summary. The flowchart of the training model is depicted in Figure 1(a) and the summarisation model is represented in Figure 1(b). The training part consists of dataset preparation. The major components of the architecture are a multilayer convolutional neural network which act as the lower level feature extractor and a stacked bidirectional GRU that extracts higher level representations from the input sequence. The role of the aggregated model is to fetch frame level features from CNN and convert it into sequence level representations for identifying the key sequences from the video.

4.1 | Video dataset preparation

The focus of this section is to describe how the videos are preprocessed for better performance before being fed to the deep learning model. The pre-processing phase begins with splitting the dataset into train, test and validation sets in the ratio 60:20:20. Further, the individual videos undergo down-sampling to eliminate redundancy. However, conventional down sampling methods based on fixed frame rate is not preferred in the proposed method because the videos under consideration are of very short duration typically ranging from 1 to 5 s and the resulting down-sampled videos may be of varying size which is not suitable for the proposed method.

Since all the videos are of duration between one and five seconds, it is feasible to fix a constant sequence length for the down-sampled videos so that the resulting down-sampled videos is of equal lengths and is the best fit for feeding to a neural network. The sequence length is set as 24 after numerous experiments by varying the sequence length values in between 15 and 25. A sampling factor is computed as the ratio of the original video length and the sequence length that is required. The sampling factor is essential for down-sampling the video as one frame from each sampling factor and the number of frames is considered for forming the video sequence. It is noteworthy that compared to conventional down-sampling methods where the sampling factor is a fixed value probably the frame rate, the proposed model uses a variable value for sampling in order to ensure that the resulting video sequences are of equal length. The feasibility of this approach is only because of the short duration of the video. The sampling factor, $\beta$ is computed as in (1).

$$\beta = \left\lfloor \frac{V_t}{S_t} \right\rfloor$$  \hspace{1cm} (1)

where $V_t$ and $S_t$ corresponds to the length of the video and sequence respectively. Once the down-sampled sequences are generated, video frames undergo pre-processing before passing them to the convolutional neural network.

4.1.1 | Frame-level pre-processing

All the video frames undergo pre-processing which involves two major phases (1) resizing and (2) Global standardisation. All the frames are processed as colour images and resized to the size 224*224 for passing to the first stage. Once the images are resized, global standardisation is applied as in (2).

$$x_i = \frac{x_i - \mu}{\sigma}$$  \hspace{1cm} (2)

where, $x_i$ is the $i^{th}$ image pixel value and $\mu$ and $\sigma$ stands for mean and standard deviation respectively. This results in the image values to have mean and standard deviation values of 0 and one respectively. However, the presence of negative values might alter the learning process. Hence, positive standardisation is applied to ensure that the image values are positive and lies in between 0 and 1. Positive standardisation is applied by initially clamping the data values to be between $-1$ and 1 and then by applying equation (3)

$$x_i = \frac{x_i + 1}{2}$$  \hspace{1cm} (3)

This guarantees that the data values lie in between 0 and 1 and have mean and standard deviation in between 0 and 1. The consistency of the input data is ensured by this which further improves the learning process.
4.1.2 | Custom video data generator with augmentation

From the pre-processed down-sampled sequences, video clips are generated from each sequence for better learning performance as shown in the Figure 1(a). Each video sequence is divided into chunks with number of frames equal to the length of the clip. From each chunk, frames from corresponding positions are combined to form clips of equal length. This will decrease the variance in the clips generated as the clips formed are uniformly spanned across the sequences. Suppose a video sequence is represented as \( V = \{f_1, f_2, ..., f_n\} \) where \( n \) is the length of the sequence. If the length of clip is 5, then the first clip can be represented by \( C_1 = \{f_1, f_6, f_{11}, ...\} \) and second clip as \( C_2 = \{f_2, f_7, f_{12}, ...\} \) and so on.

Video data augmentation is a powerful technique to improve the performance of the model. It helps in avoiding overfitting and performs better on unseen data. Augmented video clips are generated from the extracted video clips using a set of image transformations. The transformations implemented in the proposed model are rotations in different angles, horizontal and vertical flipping, and zooming. The transforms corresponding to the above augmentation techniques are applied on each frame of a video clip respectively in an orderly fashion. Rotation is performed in four random angles without considering the rotation that results while flipping horizontally.

**FIGURE 1** The Proposed framework (a) Training model (b) Summarisation model
and vertically. Variation in brightness is performed in three computations. On the whole, nine clips are generated from a single video clip which results in an increase of training dataset by 10 times. The augmentation is performed only on 80% of the dataset. Remaining 20% is considered for testing. The proposed model showed an amazing performance upgrade after employing data augmentation.

4.2 | Stage 1: multilayer CNN model

The purpose of the first stage in the proposed deep learning model is to extract frame-level features from surveillance video sequences. These frame-level features act as an input to the second stage of the model which is responsible for finding the hidden sequence level representation from the video sequence. Since the input is the down-sampled video represented as the set \( X = \{x_1, x_2, x_3, ..., x_n\} \) where \( n \) is the number of down-sampled frames and the output is a representation vector \( F = \{f_1, f_2, f_3, ..., f_n\} \) of the individual frames, the best choice for this stage would be a Convolutional Neural Network. It is worth mentioning that, since the actual classification is done in the latter stage, the accuracy of sequence level representations has higher priority than the frame level representations generated by the CNNs. Hence, the model prefers a simple CNN architecture for the first stage over the complex pre-trained CNNs that are found to exhibit higher accuracy levels. As a result, a hierarchical multilayer CNN with dual convolutions is employed, following the intuition of VGGNet architecture [29] owing to its excellent performance in extracting image features successfully. The architecture employed is depicted in Figure 2. The input frame passes through a set of dual convolutions with the number of filters being 32, 64 and 128 respectively with intermediate ReLU activations as in (4).

\[
R(x_j) = \max(0, x_j)
\]  

where, \( R \) is the ReLU function and \( x_j \) is the value of \( j^{th} \) pixel of \( i^{th} \) frame. The filter size used in the proposed CNN is 3*3. Each dual convolution is followed by a max-pool layer with pool size and stride value of 2. The max-pool layers help eliminating unnecessary information thereby retaining the necessary ones. The output from the final max-pool layer is flattened to convert the feature representation into a one-dimensional array that is fed into the second stage.

4.3 | Stage 2: stacked Bidirectional GRU model

In order to classify a video sequence as relevant or irrelevant to the summary, the frame-level features produced from the previous stage need to be encoded as sequence level features. The proposed framework identifies five most common classes of events as relevant to surveillance videos. The events that the proposed model is trained to identify are hitting, running, kicking, shooting and punching [7, 8]. These actions can be stated as undesirable events which are frequently sought in surveillance videos. In order to generate the final summary, the model is trained to identify the above-mentioned events from a surveillance video. The expected output from this stage is the probability that a sequence belongs to the output classes \( Y = \{y_1, y_2, y_3, y_4, y_5\} \). Since temporal information needs to be exploited in this stage, recurrent neural networks are the most suitable. The idea of employing various forms of RNNs in video analytics is not new. Several works in the literature have addressed the relevance of utilising unidirectional and bidirectional LSTMs for extracting long term dependencies from videos. For instance, while Cui et al. [30] has employed deep stacked bidirectional LSTMs for traffic prediction, Lee et al [31] has employed a combination of both the unidirectional and bidirectional LSTMs for anomaly detection from vibration sensor signal patterns in rotating machinery diagnostics. Consequently, specialised RNNs such as LSTMs or GRUs capable of learning long term dependencies are ideal for this stage in the proposed model. Experiments are conducted employing different types of RNNs of which the results of stacked bidirectional GRUs found to outperform the other variants. The results of the various experiments conducted are detailed in the results section 5.1.3.

4.3.1 | Bidirectional GRUs

Since RNNs fail to capture long-term dependencies due to the vanishing and exploding gradients problem, LSTMs were introduced by Hochreiter and Schmidhuber [32] which consisted of advanced recurrent unit connections aided by input, output and forget gates. A similar variant known as the Gated Recurrent Units were introduced by Cho et al. [33] which performs similar to LSTMs but is less complex than LSTMs. The major difference is that GRUs have only reset and update gates and do not possess a memory unit which makes it more efficient and at the same time unsuitable for certain applications which
require long-term dependencies. The proposed model exhibited better performance by incorporating GRUs over LSTMs. The fundamental equations that govern GRUs are stated in (5-8).

\[
z = \sigma(x_t U^z + s_{t-1} W^z) \\
r = \sigma(x_t U^r + s_{t-1} W^r) \\
b = \tanh(x_t U^b + (s_{t-1} r) W^b) \\
s_t = (1 - z) o b + z s_{t-1}
\]

where, \(z\) and \(r\) represent the update and reset gates respectively, \(b\) and \(s_t\) represent the intermediate memory and output respectively and and \(\tanh\) represent the activation functions. As the name suggests, in bidirectional RNNs the input sequence is considered as two separate sequences, one in the forward direction and other in the backward direction. The dependencies in both the directions are considered for learning and produce better results in most cases. The outputs in both directions need to be merged for further processing. The proposed architecture uses concatenate merge mode. The model performed better by incorporating bidirectional GRUs over unidirectional ones. The general structure of a bidirectional GRU is shown in Figure 3.

### 4.3.2 Stacked bidirectional GRUs

As mentioned above, the proposed framework uses three layers of bidirectional GRUs for identifying the hidden sequence representation from the video clip and for successfully classifying them. The output from stage 1 is passed through three bidirectional GRUs of 50 units each. The output sequence from second layer is passed through a third bidirectional GRU of 50 units which does not return a sequence. The merge mode of the bi directional GRUs in both layers is concatenation. Once the hidden level representation is computed by passing through this stacked bi-directional GRUs, the output is passed through a dense layer with the number of units equal to the number of classes that is trained in the proposed method. Here, the number of classes is 5, hence the dense layer will output five probability values one corresponding to the probability of a video segment belonging to that class. The activation function used for the network final layer is softmax and is computed as in (9).

\[
\hat{y}_i = \frac{e^{a_i}}{\sum_{j=0}^{N} e^{a_j}}
\]

where \(N\) is the number of classes and \(a_i\) is the \(i^{th}\) output value.

#### 4.3.3 Loss function

Since the model tries to classify the video sequences, and at this stage it is a multiclass classification problem, the perfect choice for loss function is the categorical cross entropy or log loss. Categorical cross-entropy compares the predicted probability values with the true set of values, where the probability of the true class is one and is 0 for the other classes. The true class is represented as a one-hot encoded vector, and the closer the prediction of the model are to the one hot vector, the lower the loss. The loss function is computed as in (10).

\[
H(y, \hat{y}) = \frac{1}{N} \left( \sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) \right)
\]

where \(\hat{y}_i\) is the predicted value, \(y_i\) is the true value and \(N\) is the number of samples.

#### 4.3.4 Regularisation

Regularisation helps in avoiding the overfitting of the model. It prevents the model from showing high performance on the training data and very low performance on unseen test data. The regularisation techniques generally applied are L1 regularisation, L2 regularisation, dropout, data augmentation and early stopping. The proposed model applies all but L1 regularisation to avoid overfitting.

**L2 regularisation:** L1 and L2 are the most common types of regularisation used in deep learning. These update the general loss function by adding another term known as the regularisation term due to which the values of weight matrices decrease. The optimisation function can then be stated as in (11).

\[
Cost = H(y, \hat{y}) + \lambda \sum_{i=1}^{m} \| w_i \|
\]
where, $m$ is the number of classes and is the regularisation parameter. It is the hyperparameter whose value is optimised for better results. In the proposed framework, L2 regularisation is applied in the final output layer with lambda value 0.001.

**Dropout:** Dropout is the most effective type of regularisation techniques. At every iteration, it randomly selects some nodes and removes them along with all of their incoming and outgoing connections. This results in a different set of outputs in each iteration. The proposed model applies a dropout of 0.5 immediately after the max-pool layers in the CNN. In the second stage, a recurrent dropout of 0.5 is applied in all the bidirectional GRU layers. Recurrent dropout masks the connections between the recurrent units.

**Early stopping:** Early stopping is a type of regularisation performed by stopping the training of the model when the performance on the validation set starts to decrease. The proposed framework implements early stopping by monitoring for minimum validation loss and with a patience factor of 50 epochs.

**Data Augmentation:** Increasing the size of the datasets is a definite step towards improving the performance of a network. A popular technique for increasing the dataset is to augment the existing data by applying various transforms on the datasets. The proposed model implements a custom video augmentation technique which is explained in section 4.4.

### 4.3.5 Optimization

After several experiments, the optimization technique that resulted in better convergence for the proposed model is stochastic gradient descent (SGD). The initial learning rate used is 0.01 and the decay factor is set to 0.001 and a momentum of 0.6.

### 4.4 Block-based test surveillance video data preparation

The video data preparation detailed in section 4.1 is for the training data which is not collected from actual surveillance videos. A considerable amount of the video data for training are from action dataset along with manually collected event videos. The videos in the dataset contain close up views of the events or actions. However, when it comes to real surveillance videos, the footage is typically recorded from longer distances and the background remains stationary. Hence, for identifying the presence of unusual events from the real-time surveillance videos, extracting a set of block-based video clips from the original video clip is proposed. The original video initially undergoes conventional down-sampling with frame-rate value as the criteria. Further, the video clip is formed by dividing the video into non-overlapping clips of length 5. This is to ensure that no events are missed during the analysis. Suppose the down-sampled video sequence is represented as $V$ the video clips generated can be represented as $c_{ip}$ as in (12)

$$c_{ip} = \{X_{ij}|i = 0, 1, 2, ..n and j = 0, 1, 2, ..clipLength\}$$  \hspace{1cm} (12)

where, $n$ is the number of clips generated from a video and $clipLength$ is the number of frames in a clip. According to the proposed block-based approach, each video clip results in four additional clips one from each block. The process of slicing the frames into blocks is identical to zooming onto each corner of the frame for the presence of unusual events. The block-based clips from the original clip can be represented by $c_{ipm}$ where $m = 1 \; \text{to} \; 4$ as the number of block-based clips generated per video clip is 4. The individual clip can be represented as in equation (13).

$$c_{ipm} = \{[X_{ij}]_m | m = 1, 2, 3, 4, i = 1, 2, ..n, j = 1, 2, ..l\}$$  \hspace{1cm} (13)

where, $X_{ij}$ is the $j^{th}$ frame in the $i^{th}$ clip, $m$ represents the $m^{th}$ block from the frame, $c_{ipm}$ represents the video clip corresponding to the $m^{th}$ block generated from $c_{ip}$, $n$ and $l$ are the number of clips and length of clip respectively.

### 4.5 Relevance computation and summary generation

The focus of this phase is to identify the clips that are key to the video summary. The video clips that are identified as belonging to the aforementioned classes are considered as key segment for final summary. In order to compute the probability of a clip belonging to $class_k$, first the network is initialised with the pretrained weights and bias values. Later, the probability of $c_{ipm}$ to belong to $class_k$ is computed as in (14).

$$f(c_{ipm}) = \text{prob}_{class_k}(c_{ipm})$$  \hspace{1cm} (14)

$c_{ip}$ can be a candidate segment to the summary $S$, only if the condition in (15) holds.

$$c_{ip} \in S \iff \max_m \left(\sum \text{Prob}_{class_k}(c_{ipm})\right) > \tau$$  \hspace{1cm} (15)

where, $c_{ipm}$ is the block-based clip corresponding to $i^{th}$ clip, $\text{Prob}_{class_k}$ is the probability predicted by the model that $c_{ipm}$ belongs to $class_k$, $k$ is the number of classes, $\tau$ is the threshold which is computed as the sum of mean and standard deviation of the sum of probability values from block based clips. The equation computes the sum of probabilities that a set of block-based clips from $c_{ip}$ represented as $c_{ipm}$ belong to each event $class_k$, and if the maximum of the sum of probabilities is
greater than threshold $\tau$, then $clip_i$ belongs to $class_k$ and can be a key segment in the final summary. The block-based video clip generation part in Figure 1(b) depicts how the block-based clips are generated from a video clip. Figure 5 shows the sample sequences detected from surveillance videos as key sequences for summary. The red box indicates an instance of the block-based clip corresponding to which the event is detected. Even with the minimal classes of events trained, the model is capable of successfully detecting other common suspicious events in surveillance videos that are identical to the trained events such as pushing, stomping, weapon hitting, etc. The detailed algorithms of training the model and summarising surveillance videos can be seen in Algorithms 1 and 2.

**Algorithm 1**: Supervised training of the CNN-BiGRU3 model

**Input**: Event video dataset belonging to various classes

**Output**: Trained model with weights and bias

for each video $V_i$ in each $class_k$,
- Down-sample $V_i$ with sampling factor, $\beta$
- Apply standardisation, $X_i = \frac{X_i - \mu}{\sigma}$
- Generate video clips from down-sampled $V_i$
- For each $clip_i$, generate custom augmented clips

for each epoch,
- for each clip generated $clip_i$,
  - Input frame $X_j \in clip_i$ to multilayer CNN, $j = 1$ to $cliplength$ and $i = 1$ to numberofclips
  - $\{F_j\} =$ frame-level features from $clip_i$
  - Input $\{F_j\}$ to BiGRU-3
  - Extract sequence-level features from $\{F_i\}$
  - Compute Softmax output $\hat{y}_i = \frac{e^{y_i}}{\sum_{i=0}^{n} e^{y_i}}$
  - $Loss = -\frac{1}{N} \sum_{i=1}^{N} y_i . \log(\hat{y}_i) + \lambda \sum_{k=1}^{K} ||w||^2$
- Update parameters using Stochastic Gradient Descent

**Algorithm 2**: Block-based surveillance video clip relevance computation and summarisation

**Input**: Surveillance video $V_z$

**Output**: Surveillance video summary $S$

Initialise network with pretrained weights and bias.

Divide $V_z$ into non-overlapping clips where, $clip_i = \{X_{ij}\}$ $i = 0, 1, 2, \ldots n$ and $j = 0, 1, 2, \ldots cliplength$

for each $clip_i$ form block-based clips, $clip_{im} = \{X_{ij}\}$ $m = 1 \text{ to } 4$, $i = 1 \text{ to } n$, $j = 1 \text{ to } cliplength$

Input $clip_{im}$ to the initialised model

$$f(clip_{im}) = Prob_{class_k}(clip_{im})$$

Key segment identification and summary generation

$$clip_i \in S \iff \max_{m} \sum_{i} Prob_{class_k}(clip_{im}) > \tau$$

Final Summary = \{S\}

5 | RESULTS AND ANALYSIS

Experiments are conducted in two phases; first for the optimal architecture selection and second for the performance analysis of the summarisation model. Each section begins with the datasets used followed by detailed experimental results and performance analysis.

5.1 | Experiment 1: Optimal Architecture selection

5.1.1 | Datasets used

The dataset used for training is a collection of event video shots from five classes of the HMDB51 dataset [34] in combination with custom collected event videos. Hmdb51 is a challenging data set of action classes with larger variations in background illumination, camera motion and view point and object appearance. Event detection trained on HDMI50 are found to perform consistently better. More videos related to each event class are also collected separately and is used for training.

5.1.2 | Performance metrics

The prominent metrics used in classification problems for assessing the performance of a model are accuracy, loss, precision, recall, f1-score, area under Receiver operating characteristic curve (AUC-ROC) and average precision. Accuracy measures the fraction of total number of true results retrieved by the model from the total samples considered and is computed as in (16).

$$\text{Accuracy} = \frac{\text{True Positive}}{\text{Total Samples}}$$

where Total samples (TS) is the sum of True Positives (TP), True Negatives (TN), False Positives (FP) and False negatives (FN).

Loss is the categorical cross entropy loss, also known as log loss, which is the optimization objective of the model. It computes the confidence of the prediction made by the model by computing the deviation from true value and is computed as in (7). The ideal value should be minimum and near to 0 whereas for all other metrics, the ideal value is always higher the best. Precision is defined as the fraction of relevant results in the retrieved results computed as in their equation. Recall can be stated as the relevant results retrieved from actual set of relevant results which is computed as in equation. F1-score is the most commonly used evaluation metric in classification as
it tends to balance the trade-offs in precision and recall. A good F1-score ensures that the model retrieves correct results as well as maximum correct results are retrieved. These are computed as in (17), (18) and (19).

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (17)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (18)
\]

\[
\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)
\]

Average precision gives the value of average precision at all thresholds, not just limited to thresholding at 0.5. It is similar to the area under the precision-recall curve; higher the value more ideal the model. Area under ROC curve is another important metric used to evaluate the performance of the model. ROC curve is plotted as the curve between Recall or sensitivity to False positive rate (FPR) which is (1-Specificity) or the proportion of incorrect classification made by the model as in (20).

\[
1 - \text{specificity} = \text{FPR} = \frac{FP}{TN + FP} \quad (20)
\]

5.1.3 | Experimental analysis

Comprehensive experiments are conducted for identifying the optimal architecture for the proposed framework. The various RNNs experimented begins with the basic single layer LSTM and GRU. Although LSTMs have wide acceptability and is being successfully employed for various sequence-based problems, lately, it has been found that GRUs also exhibit similar performance with lesser time and space complexity. The results recorded in Table 1 also demonstrate that LSTMs and GRUs show similar performance. However, for the proposed model, GRUs showed slightly better results along with lesser time complexity. Hence, further experiments are conducted with GRU as the base model.

Additionally, experiments performed by stacking GRU layers showed modest performance improvements over the basic ones. However, once Bidirectional GRUs were employed, there was significant boost in the performance. Experiments are further conducted by stacking bidirectional GRUs and the train and test loss and accuracy obtained by the various architectures are recorded in Table 1. Experiments with skip connections that follow the notion of residual connections were also conducted on the GRU part. The optimal values are shown in boldface. It can be observed that the values corresponding to the three layered stacked bidirectional GRUs show optimal results with an additional spike in the test accuracy of 5% with data augmentation.

Table 2 shows the performance of the various architectures in terms of the various evaluation metrics defined above. The best results are shown in boldface. It is quite evident that the results obtained by employing three stacked layers of bidirectional GRUs found to outperform all the other models in terms of accuracy, loss, precision, recall, F1-score, Area under the ROC curve and average precision. The ROC curve illustrating the performance of the various bidirectional GRUs are plotted in Figure 4. The plot is again an indication marking that the use of three layered stacked Bidirectional GRUs with data augmentation manifests finer results (0.97) in the proposed application. The area under the curve is labelled in the graph. With data augmentation, an increase in 0.06 can be noted from the plot. The spike in performance after data augmentation once again proves that data augmentation has aided in improving the performance of the proposed architecture. Although, skip connections have proven to produce reliable results in the literature, it can be observed that the proposed model did not exhibit better performance in the presence of skip connections in the GRU part. This might be due to the fact that skip connections tend to exhibit better performance on deeper networks. Since bidirectional versions of RNNs generally result in complex networks than their unidirectional counterparts, the proposed model has limited the experiments to five layers and moreover, the model was found to produce reliable results in this range. Based on the above results, the three layered stacked bidirectional GRUs with data augmentation is chosen as the optimal architecture.

5.2 | Experiment 2: summary evaluation

5.2.1 | Datasets

The datasets used for testing summarisation performance on surveillance videos are two custom datasets named ‘Street’ and ‘Campus’ collected from the YouTube dataset with videos consisting of currently 33 public streets and 22 campus surveillance videos respectively. The videos in the dataset are a diverse collection with varying events and aspects such as brightness, camera and person viewpoint, backgrounds, etc. which make them an ideal choice for performance evaluation rather than the long surveillance videos with identical sequences. The ground truth is annotated by users by manually verifying the clips from test videos and assigning them to corresponding event classes. The average length of videos are 1.95 min for Streets dataset and 1.16 min for Campus dataset.

5.2.2 | Quantitative evaluation

The summarisation model is evaluated quantitatively based on precision, recall, F1-score, computational time and compression rate. The same metrics were used in the earlier experiment for optimal architecture selection. However, in the context of summarisation, an exact match or true positive results from comparison with ground truth values. Hence, a slightly modified precision and recall can be computed as in (21) and (22) respectively where GT is ground truth summary and SS is the Surveillance video summary generated by the proposed model. F1-score is the harmonic mean as computed from (19).
Precision $= \frac{| \{ \text{GT} \} \cap \{ \text{SS} \} |}{| \{ \text{SS} \} |}$ \hspace{1cm} (21)

Recall $= \frac{| \{ \text{GT} \} \cap \{ \text{SS} \} |}{| \{ \text{GT} \} |}$ \hspace{1cm} (22)

Computation time is defined as the average time utilised by the proposed model for summary generation. Compression rate (CR) is defined as the rate of compression achieved in final summary compared to the original number of frames as in (23).

$$\text{CR} = 1 - \frac{\text{Number of key frames}}{\text{Total number of frames in the video}}$$ \hspace{1cm} (23)

Table 3 shows the precision, recall and f1-score values along with the computational time taken and compression rate obtained for the proposed model on Streets and Campus dataset along with the average values for each metric.

The model achieves elevated values of 90.3%, 69.9% and 78.54% for precision, recall and f1-score respectively. The variation in the performance on both datasets can be better observed from the plot in Figure 6(a). It is evident that the results achieved on the Streets dataset is outstanding. The drop in recall can be attributed to the fact that, although most common abnormal events are addressed in the framework, the model does not exhaustively detect all the suspicious events. Despite this fact, the model is capable of identifying events that are similar to the trained events such as stabbing, pushing, etc. On the other hand, although the performance of the model on Campus dataset is promising, the dip in recall and thereby f1-score (74.36%), owes to the fact that the model fails to identify certain campus specific abnormal events such as security breach, fire, etc. that have an equal rate of occurrence.

### Table 1
Comparison of accuracy and loss of train and test set for the various architectures experimented

| Model             | Train set | Test set |
|-------------------|-----------|----------|
|                   | Loss      | Accuracy | Loss      | Accuracy |
| LSTM              | 0.22      | 0.88     | 1.10      | 0.56     |
| GRU               | 0.20      | 0.90     | 1.30      | 0.55     |
| Stacked GRU       | 1.12      | 0.90     | 0.33      | 0.59     |
| BiGRU-1           | 1.34      | 0.93     | 0.20      | 0.65     |
| Stacked BiGRU-2   | 0.34      | 1.00     | 0.09      | 0.77     |
| Stacked BiGRU-3   | **0.01**  | **1.00** | **0.19**  | **0.82** |
| Stacked BiGRU-4   | 1.10      | 0.83     | 1.12      | 0.67     |
| Stacked BiGRU-4 with skip connections | 0.88 | 0.91 | 1.67 | 0.62 |
| Stacked BiGRU-5   | 1.14      | 0.76     | 1.25      | 0.63     |
| Stacked BiGRU-3 with data augmentation | **0.19** | **1.00** | **0.23** | **0.87** |
| Stacked BiGRU-3 with skip connection and DA | 0.02 | 1.00 | 0.47 | 0.73 |

### Table 2
Comparison of the performance of the various models based on classification metrics

| Model                      | Performance Measures |
|----------------------------|-----------------------|
|                            | Precision | Recall | F1-score | AUC-ROC | AP  |
| GRU                        | 0.65       | 0.64   | 0.64     | 0.63    | 0.6  |
| LSTM                       | 0.77       | 0.56   | 0.65     | 0.66    | 0.73 |
| Stacked GRU                | 0.65       | 0.7    | 0.67     | 0.66    | 0.63 |
| BiGRU-1                    | 0.62       | 0.60   | 0.60     | 0.73    | 0.58 |
| Stacked BiGRU-2            | 0.70       | 0.67   | 0.68     | 0.80    | 0.67 |
| Stacked BiGRU-3            | **0.81**   | **0.79** | **0.80** | **0.91** | **0.83** |
| Stacked BiGRU-4            | 0.57       | 0.56   | 0.57     | 0.69    | 0.52 |
| Stacked BiGRU-4 with skip connections | 0.48 | 0.47 | 0.46 | 0.62 | 0.43 |
| Stacked BiGRU-5            | 0.5        | 0.49   | 0.49     | 0.63    | 0.45 |
| Stacked BiGRU-3 with data augmentation (DA) | **0.89** | **0.88** | **0.88** | **0.97** | **0.95** |
| Stacked BiGRU-3 with skip connections and DA | 0.79 | 0.76 | 0.77 | 0.88 | 0.79 |
as the trained events in campuses compared to other public areas. The results of the average time taken to generate summaries is also noteworthy. However, it should be mentioned that the videos considered currently in the dataset are short videos which are less than 3 min. Notwithstanding, the model behaves efficiently in terms of computational time. Similarly, the results of the average compression rate show the average extent to which the videos in a dataset are compressed after summarisation, is also remarkable. The high value of compression rate can be attributed to the capability of the model to precisely detect exactly the abnormal events from the videos.

The comparison of the performance of the model on five state-of-the-art works by Gao et al. [18], Muhammad et al. [14], Muhammad et al. [16], Muhammad et al. [17], Liu et al. [19] and Lei et al. [21] based on f1-score are plotted in Figure 6(b). The results prove that the performance of the model surpasses most of the state-of-the-art works given the minimal number of classes trained and complexities involved. It is worth mentioning that the values plotted are the average obtained on both datasets. The proposed model can exhibit an even superior performance on more public surveillance datasets like Streets than surveillance footage of places with restricted access such as campuses and offices.

5.2.3 Qualitative evaluation

For assessing the summary quality of the proposed model and the baseline models, qualitative evaluation is conducted on a set of 25 users. The users are asked to score each quality parameter based on the summary generated. The quality parameters considered for evaluation are representativeness, informativeness, diversity, completeness and conciseness. Representativeness valuates the capability of the generated summary to portray the original video. Informativeness is the extent to which the generated summary can provide important aspects of the original video. Diversity assesses the summary in

**Figure 4** ROC curves of the RNN models experimented

**Figure 5** Sample key segments detected from Surveillance videos by the proposed model for summaries depicting (a) punching event (b) pushing and stomping (c) shooting with gun
TABLE 3 Performance of the proposed model based on the quantitative metrics on datasets

| Datasets    | Precision | Recall | F1-score | Computational time(seconds) | Compression Rate |
|-------------|-----------|--------|----------|----------------------------|-----------------|
| Streets     | 90.02     | 77.53  | 82.72    | 12.17                      | 99.89           |
| Campus      | 90.67     | 64.1   | 74.36    | 10.33                      | 99.83           |
| Average     | 90.34     | 69.98  | 78.54    | 11.25                      | 99.86           |

FIGURE 6 Performance of the proposed model based on (a) Quantitative measures on the two datasets (b) state-of-the-art models

terms of variation in events or scenes which is also a measure of redundancy.

The performance of the proposed system on the Streets and Campus datasets based on the user responses are plotted in Figure 7 (a) and (b) respectively. The superiority in performance is obvious in Streets dataset whereas the performance results in Campus dataset are comparatively lesser. It can be observed that the representative factor and completeness are the parameters that are relatively lower in performance. This can be attributed to the fact that the surveillance video summary is not highly representative of the whole video owing to the length of the video and the fact that most of the time no particular events take place in the video. Since, increasingly more events need to be addressed for a complete summary, it contributes to the dip in completeness factor. The results yielded in Campus dataset is satisfactory. Although, an additional dip in informativeness can be observed. This is accountable to the fact that there are equally occurring threats as with the trained events in campuses such as unauthorised entries, drug abuse, vandalism, etc. which the model is not trained to detect. However, as per recent reports, there has been a spike in campus violence which the model is able to identify successfully with the trained model. Although, there is a slight dip in representativeness, informativeness and completeness factor, overall results prove the quality of the summaries produced by the proposed model. Further, comparison of the performance of the proposed summarisation model with the experimented baseline models on Street and Campus datasets are also provided in Figure 8 (a) and (b) respectively. It can be noted that the proposed stacked bidirectional GRU with three layers and data augmentation is the optimum one for summarisation in terms of the qualitative metrics as well, thereby, substantiating the quantitative results.

It is noteworthy that even though the diversity and conciseness factors achieved by the baseline models is satisfactory, the key factors such as representativeness, informativeness and completeness lack user score. Moreover, the conciseness factor obtained by the models further substantiate the compression rate achieved as well.

6 | CONCLUSION

The proposed framework is a multi-stage deep learning model for surveillance video summarisation based on abnormal event detection. The model is built by combining a multi-layer CNN with a stacked bidirectional Gated Recurrent Unit. Two novel datasets for surveillance video summarisation have been introduced. A comprehensive experimental analysis on the selection of the optimal model for video sequence classification is conducted. Results substantiate the significance of the three layered stacked bidirectional GRU employed for sequence level feature extraction in the proposed application. Additionally, the custom video sequence generator and augmentation techniques implemented resulted in an increase in the classification metrics scores such as F1-score, AUC and AP. A thorough analysis on the performance of the summarisation model is also performed based on quantitative and qualitative measures. Comparison with state-of-the-art models shows a comparatively elevated F1-score of 78.54%. The results further prove that the block-based approach put forth for summarising real surveillance video sequences has aided in enhancing the performance of the model. The qualitative results obtained further establish the significance of the proposed model for surveillance video summarisation. Moreover, with the minimal classes of events trained, the model could
Figure 7  Performance of the proposed model based on qualitative parameters on (a) Streets dataset (b) Campus dataset

Figure 8  Comparison of the performance of proposed models with baseline models based on qualitative parameters on (a) Streets dataset (b) Campus dataset
successfully identify similar suspicious acts and resulted in an enhanced performance in the summarisation phase. Future works focus on improving the performance of the summarisation model by experimenting with complex encoder decoder architectures and generative adversarial networks (GAN) which, lately, have shown outstanding performances. Enhancement of the datasets to make it large-scale by including more heterogeneous video collections is a prospective future research concern. The unavailability of large annotated surveillance video datasets for supervised learning and performance evaluation is also a potential challenge that needs to be addressed in the near future.

REFERENCES

1. Baskurt, K.B., Samet, R.: Video synopsis: a survey. Comput Vis Image Underst. 181, 26–38 (2019)
2. Sreeja, M.U., Kovoor, B.C.: Towards genre-specific frameworks for video summarisation: a survey. J Vis Commun Image Represent. 62, 340–38 (2019)
3. Thomas, S.S., Gupta, S., Subramanian, V.K.: Event detection on roads using perceptual video summarisation. IEEE Trans Intell Transp Syst. 19(9), 2944–54 (2017)
4. Athanasiou, J., et al.: Detecting abnormal events in traffic video surveillance using super-Orientation optical flow feature. IET Image Processing. 14(9), 1881–1891 (2020)
5. Thomas, S.S., Gupta, S., Subramanian, V.K.: Smart surveillance based on video summarisation. In: IEEE region 10 Symposium (TENSYMP), pp. 1–5 (July 2017)
6. Kumar, K., Shitmanlar, D.D.: F-DES: Fast and deep event summarisation. IEEE Trans. Multimedia. 20(2), 323–34 (2017)
7. Sohianhi, F., Izquierdo, E., Piatr, T.: Ontology-based forensic event detection using inference rules. Int. Conf. Engg. Tech. Innovation, 584–591 (Jun 2017)
8. Kahr, N.F., Izquierdo, E.: Ontology-based analysis of CCTV data. In: Proc. Latin American Conf. On networked and Electronic Media, Valparaiso, pp. 62–67 (2017)
9. Nasir, M., et al.: Fog computing enabled cost-effective distributed summarisation of surveillance videos for smart cities. J Parallel Distr Com. 126, 161–70 (2019)
10. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarisation with adversarial lstm networks. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 202–211 (2017)
11. Panagiotakis, C., Papadakis, H., Fragopoulou, P.: Personalised video Summarisation based Exclusively on user Preferences. In: European Conference on Information Retrieval, pp. 305–311. Springer, Cham (April 2020)
12. Rochan, M., Ye, L., Wang, Y.: Video summarisation using fully convolutional sequence networks. Proc. Compt. Vis. ECCV, 347–363 (2018)
13. Zhao, B., Li, X., Lu, X.: Hsa-rnn: hierarchical structure-adaptive rnn for video summarisation. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 7405–7414 (2018)
14. Muhammad, K., Hussain, T., Baik, S.W.: Efficient CNN based summarisation of surveillance videos for resource-constrained devices. Pattern Recognit. Lett. 130, 370–375 (2020)
15. Zhang, Y., et al.: Unsupervised object-level video summarisation with online motion auto-encoder. Pattern Recognit. Lett. 130, 376–385 (2020)
16. Muhammad, K., et al.: Cost-effective video summarisation using deep CNN with hierarchical weighted fusion for IoT surveillance networks. IEEE Intern. Things J. 7(5), 4455–4463 (2020)