International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 01, Issue 02, November – December (2023)

RESEARCH ARTICLE

DETECTION OF VIOLENCE IN FOOTBALL STADIUM THROUGH BIG DATA FRAMEWORK AND DEEP LEARNING APPROACH

M. Dhipa[1, *] and D. Anitha[2]

[1]Associate Professor, Department of Biomedical Engineering, Nandha Engineering College, Erode 638052 India.
[2]Associate Professor, Department of Information Technology, Muthayammal Engineering College, Namakkal, Tamil Nadu, India.

**Corresponding e-mail: dhipa02m@outlook.com*

Abstract – Football is the most famous game in the world, with over 4 billion supporters worldwide. Football hooliganism refers to the aggressive or destructive actions of a supporter or player in a stadium while watching or participating in a game. To avoid violence, a real-time violence detection system is required to observe the audience and players behaviour in order to take appropriate action before violence occurs. The input of the system is a massive volume of real-time video feeds from various sources, that are processed using the Flink structure. Using the Histogram of Oriented Gradients (HOG) function in the Flink framework, pictures are partitioned into frames and their characteristics are retrieved. The frames are then labelled on the basis of attributes including Groundside-violence model, Crowdside-violence model, human part model, and Non-violence model, are utilised to train the multihead attention based Bidirectional Long Short-Term Memory network for violent scene detection. The RWF-2000 dataset, which contains the training set (80%) and the test set (20%) was used to train the network and also a dataset comprising 410 video footages with non-violence scenes and 409 video footages with violent situations is created by the videos obtained from a football stadium, to make the algorithm more strong to violence detection. Other existing approaches are used to validate the model's performance. When compared with existing systems, the proposed violence detection methodology significantly increases accuracy upto 1.6453%, precision upto 0.646%, recall upto1.959%, and reduces execution time upto 60% than other existing methods.

Keywords *– Multi-head attention, LSTM, Bidirectional-LSTM, RWF-2000, violence detection, Flink framework*

1. INTRODUCTION

With the expansion and improvements in the domain of computer vision over the last decade, a huge amount of current approaches have arisen and enticed a lot of attention from researchers owing to their wide range of surveillance applications [1]. Video surveillance is crucial in detecting human behaviour and preventing or reducing violence in real time [2]. Violence detection is a critical step to identify regular human activities from abnormal/violent acts in the development of automated security surveillance systems. Regular life interacting actions, such as walking, hand waving, running, and jogging, are frequently classified as normal human activities. Violence, on the other hand, is exposed to unusually ferocious activities, such as fights involving two or more persons [3].

Surveillance using complete human operators has also shown to have several flaws, such as high staffing costs, variability in long-duration capture, and poor multi-screen monitoring capability [4]. The identification and tracking of moving object are the key problems in computer vision-based surveillance systems [9]. Moving object detection is the technique of recognizing a thing in an audiovisual that is shifting its position in relation to the scene's turf of opinion [5]. Hence, an automatic violence detection system is needed [4]. As a result, big data analytics employing deep learning approaches may be able to identify violence more quickly and accurately.

A type of AI and ML that mimics the learning process of humans is deep learning. Deep learning has been more significant in big data analytic solutions in recent years [6]. Big data [8] is an area which involves ways for testing, consistently extracting data from, or deals with data volumes which are overlarge or complicate for traditional data-processing application software to manage. Deep Learning [10] is a strong technique for Big Data Analytics because it can analyse and learn massive amounts of unsupervised data [7]. The core goal of immense statistics analytics is to identify valuable models from large amounts of information which could be utilized for decision-making and prediction [8].

Football matches are one of the most popular types of entertainment, yet they are frequently disrupted by violence between fans or between players. The present difficulty of violence detection in sports data processing is that it is difficult to obtain significant information for classification and notifying security personnel in a short time. Conventional operators viewing surveillance footage respond slowly that results in loss of human life and property. Hence an automated violence detection system which works in a short duration is needed.

This paper proposes a technnique for violation detection system in the football ground that alerts the security persons within a short period of time. Immense statistics analytics and DL will be combined to improve the effectiveness of the violence detection system. Here, the videos from the surveillance cameras are processed using the Flink structure. By using the HOG purpose, the Flink outline divides the edges and retrieves the image frame characteristics. The frames are then labelled on the basis of attributes including Groundside-violence model, Crowdside-violence model, humanoid part model, and non-violence model, are utilized to train the MH-BLSTM for violent scene detection.

Following are the remaining sections of the paper. The literature review is represented in Section II. The proposed technique is represented in Section III, which uses the big data framework namely Apache Flink and Deep learning based Multi head -Bidirectional LSTM. The results and discussion are labelled in Section IV. The conclusion is described in Section V.

2. LITERATURE REVIEW

In 2021, Peixoto et al., [11] presented a method for allowing devices to perceive a top-level notion of violence by dividing into little, more realistic components, including fights, blood, bullets, and explosions, and then combining them afterwards for a clearer idea of the picture. Because of its robust and flexible construction, the detector may be tailored to fit the needs of a wide range of cultures and users. Experiments indicate that the presented technique performs better than existing methodologies.

In 2021, Ullah, et al., [12] provide an operative and resilient method for detecting abnormalities in BVD using AIoT. To assess the effectiveness of their methodology, they run comprehensive experiments on benchmarks constructed on peak of the RWF-2000 and UCF-Crime datasets. In comparison to existing approaches examined across the aforementioned datasets, they claim a 9.88 percent and 4.01 percent improvement in accuracy.

In 2020, Cameron, et al., [13] developed a unique strategy that utilises the object detection algorithm's past detection confidences to build the best independent prioritizing score. A new ensemble technique is also described, which employs a KNN regressor to aggregate the superior of the formerly analyzed measure to build a active prioritizing technique. Using three publicly accessible datasets, this technique is proven to boost the target identification ratio to 60% in comparison to a static sub-sampling baseline.

In 2020, Guedes, and Chávez, [14] provides a technique on the basis of Dynamic Images approach, which employs handmade and CNN features such as the Bag of Visual Words paradigm and an SVM classifier to recognise violent acts including bodily struggle in video streams from literary databases. For the Hockey dataset, the suggested approaches produce an average accuracy of 97.50 percent, 99.80 percent for the Movies dataset, and 93.40 percent for the Crowd dataset. Furthermore, each video's detection of violence was done in hundredths of a second.

In 2021, Wang et al., [15] proposed a brute force detection approach on the basis of integration of convolutional neural networks and trajectory to address the issue of unusual behaviour detection, particularly the poor efficacy and less precision of brute force detection. Using the Hockey and Crow datasets, the study's proposed brute force identification method demonstrated up to 92 percent and 97.6 percent, respectively, accuracy. Experimental data indicates that the violence detection approach proposed in this study improves the video violence detection accuracy.

In 2020, Deepak et al., [16] examined the gradient-lowed attributes' spatiotemporal autocorrelation in order to efficiently identify violent acts in crowded environments. The effectiveness of machine learning algorithms is significantly impacted by the format of the raw data. Then, to detect violent acts in videos, a discriminative classifier is utilised. On comparison with existing methodologies, experimental findings reveal that the suggested methodology outperforms them.

In 2021, Islam et al., [17] propose a 2-stream DL architecture based on Separable Convolutional LSTM (SepConvLSTM) and pre-trained MobileNet, in which 1 stream examines difference of neighbouring frames while the other stream requires in context restrained frames as inputs. On the bigger and more difficult RWF-2000 dataset, their model surpasses the accuracy by more than 2%, while corresponding results of existing method on the lesser datasets. Their results show that the suggested models outperform the competition in terms of computing efficacy and identification accuracy.

In 2018, Mumtaz et al,[18] suggested a deep representation-based model for detection vicious scenarios utilizing the notion of transfer learning to recognize violent human actions. The results indicate that the suggested approach surpasses accuracies of existing methods by learning the most discerning features, which achieve 99.97% and 99.28% accuracies on the Movies and Hockey datasets, appropriately, and by learning the best features for the activity of violent behavior identification in footages.

In 2020, Abdali, and Al-Tuma, [19] suggested a model that comprises of CNN as a spatial feature extractor and LSTM as a temporal relation learning approach with a concentration on the 3-factor model (total prevalence - correctness–quick reaction time). At a frame rate of 131 frames per second, the suggested model achieved an accuracy of 98%. The accuracy and speed of the proposed model were compared to previous research, and it was found that, out of all the approaches already in use for violence detection, the suggested technique had the highest accuracy and the fastest speed.

In 2020, Ehsan, and Mohtavipour, et al., [20] presented a unique Vi-Net architecture on the basis of a deep Convolutional Neural Network (CNN) to identify activities with unusual speed. Optical flow vectors are used to train the Vi-Net network by estimating the movement patterns of objects in the video. They conducted multiple tests on the Hockey, Crowd, and movies datasets, and the results revealed that the suggested architecture outperformed existing approaches relating to accuracy.

3. PORPOSED METHOD

The core factor of the suggested method is to detect the violence in real time for preventing the violence in advance by alerting the security personnel in a short period of time. To succeed in that a violence detection system with better performance have been proposed. Figure 1 represents the work flow of the suggested system.

An input source for the proposed system is a stream of videos from various sources. Video streams are turned into non-overlapping pictures from video streams using the Flink framework. Upon receiving the incoming video streaming block, the Flink framework converts it to frames and passes them to the HOG function in the Flink engine for feature extraction. Flink is faster than other traditional systems in processing streaming blocks in real time. Once the frames have been processed, they are divided into 8 8-pixel cells, with gradient orientations created for every cell. After normalizing the histogram employing surrounding pixels, the features are excerpted from the pictures. According to the activities, the photos are divided into four parts. The "Ground side-violence model" views the fight between the players in the ground. The “crowdside-violence model” contains the physiological harm, maldevelopment or deprivation among the crowd in the audience side. The “Human part model” has representations of autonomous human parts that have been set aside during the violence. The “non-violence model” includes visuals of a background of a specific location where there is no violence.

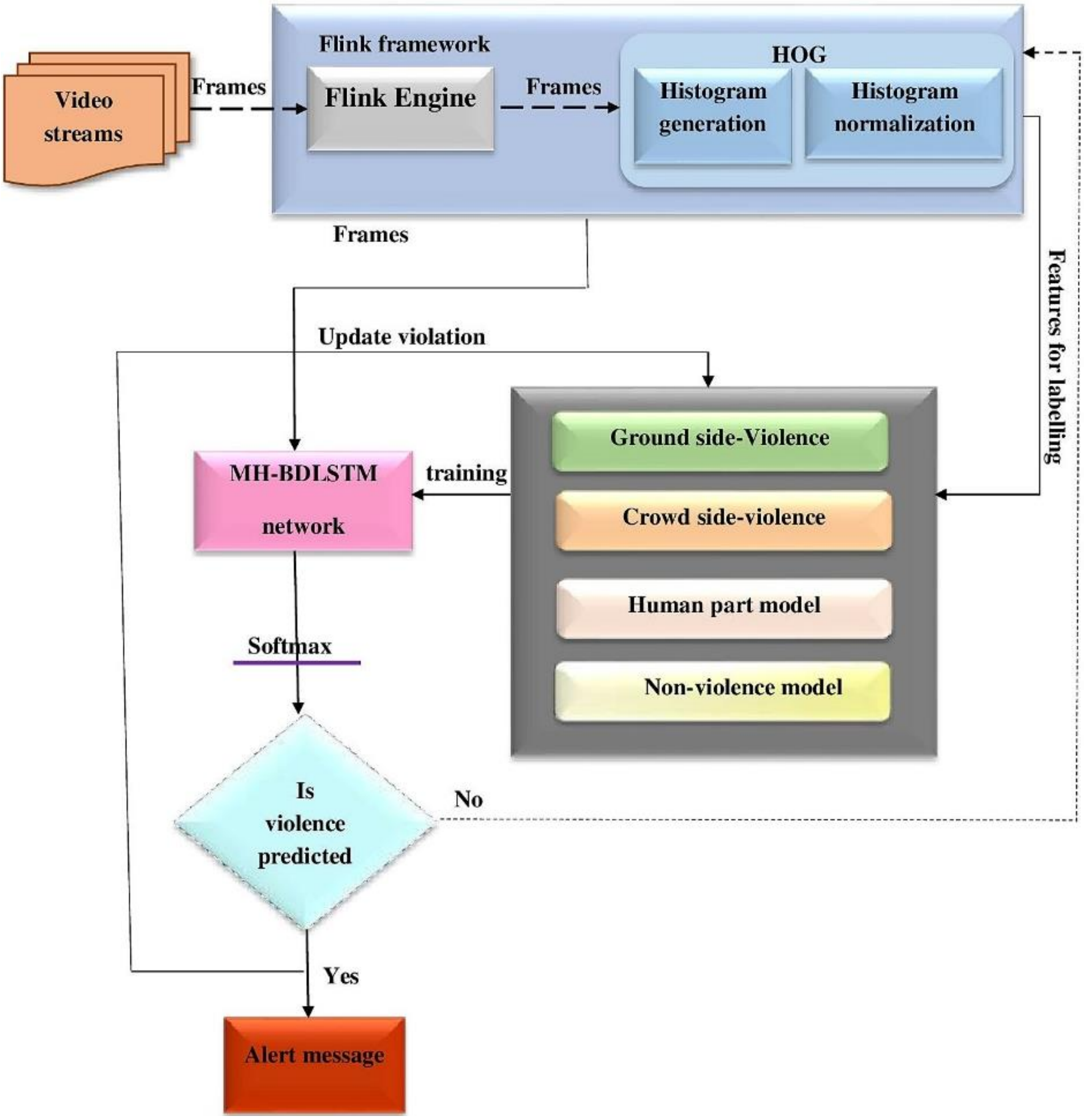


Figure 1. Architecture diagram for the proposed violence detection system

Next, the multihead Bidirectional LSTM (MH-BDLSTM) network is trained using all of the models. Following the training of the network, MH-BDLSTM validates the input frames sequentially and detects violent actions, which can handle both forward and backward dependencies. A multi-head time-dimension approach is used instead of BDLSTM outputs to obtain more valuable space - time information connected to violence detection by reflecting the outputs into other sub-images. A security officer is notified (Groundside Violence, Crowdside Violence) if the input frames match the violent models. Thus, the identified or predicted violent frames in the video are saved in the violence models for later reference. The next subsections will cover the Flink approach, the HOG algorithm, and the MH-BDLSTM neural network in detail.

3.1. Apache Flink

The Flink framework is a distributed processing engine for unbounded and bounded streams of data. Flink is intended to carry out in-memory calculations at any size in all cluster settings. Streams of data are created every time an event occurs. Flink runs any bitstream programme in a data-parallel and pipelined (thus task parallel) fashion. Flink's pipelined runtime technology enables stream processing and bulk/batch programmes to be implemented. Additionally, Flink's runtime natively allows the implementation of repetitive techniques.

Flink data streaming run-time ensures low latency and high throughput with minimum installation requirements. The failure tolerance method in Apache Flink is predicated on Chandy-Lamport distributed snapshots. Because the technique is less weight, it can sustain high throughput rates while also providing excellent consistency guarantees. Flink would’ntsupply its personalinformationrepository system, but it does provide connectors to Amazon Kinesis, HDFS, Apache Cassandra, Apache Kafka, and ElasticSearch.

Flink is the 4G of Big Data. Flink manages memory for clients automatically. Flink works with data at a rapid speed. It can connect to Hadoop and NoSQL databases natively and handle HDFS data. This is the quickest tool for evaluating big data, such as a video stream. The data stream can be processed rapidly and efficiently. Apache Flink, which is based on a pure streaming method, is used to provide good throughput and simplicity of use for sophisticated analysis processing in big data.

3.1.1. Checkpoints, Fault-Tolerance, and Save points

Apache Flink implements a less weight defect tolerance method on the basis of distributed checkpoints. Checkpoints are automatic snapshots of an application's state and a place in a stream. Flink programs with checkpointing configured resume execution from the previous completed checkpoint after a mistake, thereby guaranteeing that application state semantics are preserved. Utilizing hooks exposed by the checkpointing mechanism, external systems can be included in the checkpointing mechanism. Also included in Flink are checkpoints that are physically activated. A user can create a save point, then halt and resume a running Flink programme from the similar application state and location in the stream. Save points allow you to update a Flink programme or a Flink cluster without sacrificing the state of the application. Save points, which were introduced in Flink, now allow users to resume an application with a distinct parallelism, enabling them to adjust to shifting workloads.

Advantages

When compared to other big data technologies, it offers several advantages. The benefits are

- Flink provides APIs that are simpler to develop than MapReduce APIs. It enables for in-memory dispensation, which is significantly quicker. It also adds additional operators to the MapReduce concept, such as join, cross, and union.
- Flink is capable of analysing real-time stream data, graph processing, and machine learning methods. Faster throughput, a shorter latency, and a guarantee that exactly one processing will take place with Flink.
- Flink is also regarded as a viable replacement for Spark and Storm.

3.1.2. API for DataSets

On limited datasets, Flink's DataSet API allows for conversions. There are about 20 distinct types of conversions in the DataSet API. The Application Programming Interface is provided in Scala, Java, and a Python API that is still in development. The DataSet API in Flink is identical to the DataStream API in principle.

3.1.3. SQL and the Table API

The Table API in Flink's Scala and Java DataStream and DataSet APIs provides a SQL-like expression language for batch processing and relational stream. A relational Table abstraction is used by the Table API and SQL interface. External data sources, as well as existing DataStreams and DataSets, can be used to generate tables. On Tables, relational operations include selection, joins, and aggregation are supported via the Table API.

Regular SQL may also be used to query tables. Table API and SQL are functionally comparable and may be used together in the same software. The logical plan, that was specified by SQL queries and relational operators, is optimised employing Apache Calcite and changed into a DataSet or DataStream programme once a Table is transferred back into a DataStream or DataSet.

3.1.4. Flink streaming

Apache Flink uses the Kappa construction. The core impression overdue the Kappa design is to use a single stream processing engine to manage together group and real-time statistics. The Lambda architecture, which includes distinct processors for batch and streaming data, is used by the majority of big data frameworks. Architecture of framework shown in Figure 2.

Batch dispensation refers to the processing of large amounts of data in a single batch over a particular timeframe. Continuous streams of data can be instantly processed with stream processing. Data streams can be handled by Flink

Streaming in real-time, utilizing the pipelined Flink engine and include customizable windows. The feature extraction can be carried out employing HOG function.

3.1.5. Features of Flink

- Using the DataStream API, Flink programmes enable users to execute business logic requiring a contextual state while processing data streams at any scale, resulting in stateful streaming at any scale.
- Flink provides a defect-tolerance method on the basis of asynchronous checkpointing and periodic
- Exactly-Once Consistency: In the event of a failure, the Flink core guarantees that every event in the stream is supplied and worked precisely once.
- Scalability: programmes are parallelized so that the number of processing jobs may be increased or decreased.
- Flink apps use local, typically in-memory, state to complete all calculations, resulting in very less processing delays.
- Flink connects to a wide range of data sources, such as Elasticsearch, Apache Cassandra, Apache Kafka, Kinesis, and many others.
- Deployment options: Flink is compatible with a variety of cluster setups, including YARN, Apache Mesos, and Kubernetes.
- A library for detecting patterns in data streams using Complex Event Processing (CEP).
- Java and Scala have fluid APIs.
- Flink is a genuine streaming engine, as opposed to Spark Streaming's micro-batch processing paradigm.

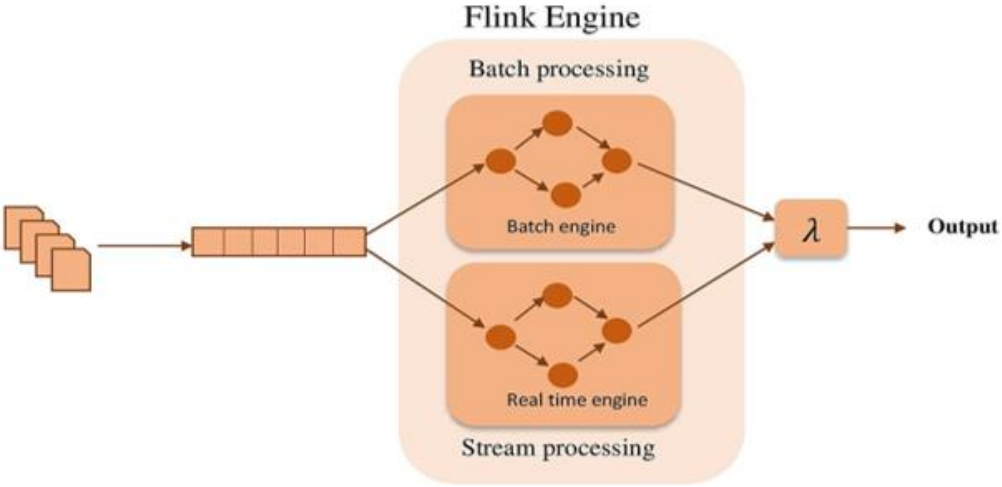


Figure 2. Architecture of Flink framework

3.2. Histogram of Oriented Gradients (HOG) function

The histogram of directed gradients is used in computer vision and image processing to identify objects. Gradient orientations are counted based on the number of times they appear in a particular section of a photograph. This approach uses overlapping local contrast normalisation and is based on a dense grid of evenly spaced cells, which sets it apart from edge orientation histograms, shape, and contextual scale-invariant feature transform descriptors.

This histogram describes the aspect and form of local objects inside an image by measuring the delivery of gradients or edge directions. After the frame is splitted into slight corresponding pieces known as cells, a histogram of gradient oeientations for each pixel is constructed. These histograms make up the descriptor. The function of HOG will be carried out in 4 stages. They are described as follows:

3.2.1. Gradient Calculation

The calculation of the gradient values is the initial stage in the process. The gradient is calculated by summing the angle and magnitude of the picture. First, the Gaand Gb values are found for each pixel in a 3x3 pixel block. Ga and Gb are initially calculated for each pixel value using the following formulas.

$$G_a(R,C) = I(R,C+1) - I(R,C-1) \tag{1}$$

$$G_b = I(R-1,C) - I(R+1,C) \tag{2}$$

where R and C stand for rows and columns, respectively. Once Ga and Gb have been established, the magnitude and angle of each pixel are computed using the following formulas.

$$Mag(\varphi) = \sqrt{G_a^2 + G_b^2} \tag{3}$$

$$Angle(\theta) = |\tan^{-1}(G_b/G_a)| \tag{4}$$

3.2.2. Orientation binning

The second step in the procedure is to create cell histograms. Each pixel in the cell represents a weighted vote for an orientation-based histogram channel, depending on the principles that were discovered during the gradient computation. The cells of a histogram can be rectangular or radial, and the channels are evenly spaced over a range of 0 to 360 degrees or 0 to 180 degrees, depending on whether the gradient is signed or unsigned. For every block, an n-point histogram is produced. Using an n-point histogram, n bins with θ-degree angle ranges are produced in the histogram.

$$Number\ of\ bins = n(ranging\ from\ 0^o\ to\ 180^o) \quad (5)$$

$$step\ size(\Delta\theta) = 180^o/Number\ of\ bins \quad (6)$$

3.2.3. Descriptor blocks

The gradient capacities must be locally adjusted to accommodate for differences in light and contrast, that requires combining the cells collectively into bigger, geographically linked blocks. Circular C-HOG blocks and Rectangular R-HOG blocks are the two most common block geometries.

3.2.4. Block normalization

A normalisation process reduces the negative impact of differences in contrast between images of the same thing. In general, the gradient is affected by lighting. When we divide or multiply pixel terms by a value to attain lighter or darker, the gradient magnitude and histogram values alter. It is important that histogram values remain unchanged regardless of lighting conditions. Within a block, the histogram vector v is normalized. One of the following standards could be suitable:

- L1 norm
- L2 norm
- L2-Hys (Lowe-style clipped L2 norm)

Let fyj be a non-normalized vector that contains all the histograms in a particular block. The L2 norm is used to standardize the fy values for each block:

L2 norm:

$$f_{yj} \leftarrow \frac{f_{yj}}{\sqrt{\|f_{yj}\|^2 + s}} \quad (7)$$

Where s is the minimum worth, additional to the square of fyj in order to evade zero separation error.

$$t = \sqrt{y_1^2 + y_2^2 + \cdots y_n^2} \quad (8)$$

$$f_{yj} = \left[\left(\frac{y_1}{t}\right), \left(\frac{y_2}{t}\right), \dots \left(\frac{y_n}{t}\right)\right] \quad (9)$$

A video stream patch is divided into eight nonoverlapping pixels, or cells, to form this study. The gradients for each pixel are determined in these cells. The surrounding cells can be used to standardize this histogram. This will increase the robustness of the texture and lighting variations. The edges are labeled in agreement with the Groundside-Violence Model, Crowdside Violence Model, Human Part Model, and Non-violence Model after the features have been extracted by employing the HOG function. The MH-BDLSTM network is then used to train the model.

3.3. Multi Head Bidirectional LSTM

For the violence detection model, a hybrid modelon the basis of bidirectional LSTM and multi-head attention mechanism was developed. The framework is depicted in the image below (Figure 3). The suggested model structure is categorized into four sections.

- Extract the text's word embedding
- Upload the word vector to the BiLSTM framework.
- Implement a Multi-head Attention method to acquire important data from several subspaces and emphasise the significance of various aspects;
- For detection of violence, use the softmax layer.

3.3.1. Word vector representation

Words are at the bottom of the structure, with a subtext X composed of n number of words being represented as (y1, y2, y3..., yn-2, yn-1, yn), and the input layer as:

$$I = [wi_1, wi_2, wi_3, \dots, wi_{m-2}, wi_{m-1}, wi_m] \in R^{m\times d} \quad (10)$$

3.3.2. BidirectionalLSTM

The LSTM is a sort of RNN which could learn and remember long-term dependencies without experiencing vanishing gradient descent or explosion difficulties. The sequence's future data is recorded in the backward layer, and its past data is kept in the forward layer. Both layers share the same output layer. The multi-head technique is used in this network, which leverages bidirectional LSTM. Multi-head attention permits the forms to concurrently assist to intake from various representation subspaces at several places.

3.3.3. Layer of word encoder

When the words are expressed using word embedding W, each word in the comment C is independent of the other words. A new representation for each word is created in this layer by combining contextual data from both directions in a comment. A bidirectional LSTM is made up of a forward LSTM H which discusses feedback from c_1 to c_n.as well as a backward LSTM $\overleftarrow{H}$that reads the comment from $c_n$toc_1.

$$\overrightarrow{H}_M = \overrightarrow{LSTM}(W_m, \overrightarrow{H_{m-1}}) \quad (11)$$

$$\overleftarrow{H_M} = \overleftarrow{LSTM}(W_m, \overleftarrow{H_{m+1}}) \quad (12)$$

To derive hidden state representation H_m for each word W_m, simply concatenate forward hidden state $\overrightarrow{H}$ and backward hidden state $\overleftarrow{H}$, i.e. H_m = [$\overrightarrow{H_m}, \overleftarrow{H_m}$]. This method aids in the capture of information from the entire sentence around each word W_m. $H \in R^{N\times 2p}$ denotes all the hidden states of the words W_m, with $\overrightarrow{H}$and $\overleftarrow{H}$ being s in size.

$$H = (H_1, H_2, \dots H_n) \quad (13)$$

3.3.4. Multihead attention

MHAT is an improved version of the classic attention mechanism that also outperforms it. Several factors can influence how a frame is paid attention, requiring the use of multiple heads of attention, in which each frame is assigned appropriate weight based on multiple aspects to convey the overall semantics of the statement. One head is calculated at a time in this method. It's important to remember that there are h times to accomplish this, that is known as multi-head, but the parameters W for every linear relationship of Q, K, and V are distinct. In the following step, all m times scaled dot-product attention outcomes are combined, and the linearly transformed value is utilised as the MHAT result.

$$H_1 = attention(QW_j^Q, KW_j^K, VW_j^V) \tag{14}$$

$$M(Q, K, V) = Concatenate(H_1, \ldots . H_n)W^P \tag{15}$$

This method explores the inner connections of sentences using self-attention, here K = V = Q. A weight matrix M and a feature representation frep are generated by this MHAT process.

$$y_i = \tan h(W_r H_i + a_r) \tag{16}$$

$$\mathrm{f_{rep}} = \mathrm{Multihead}(\mathrm{Y}, \mathrm{Y}, \mathrm{Y}) \tag{17}$$

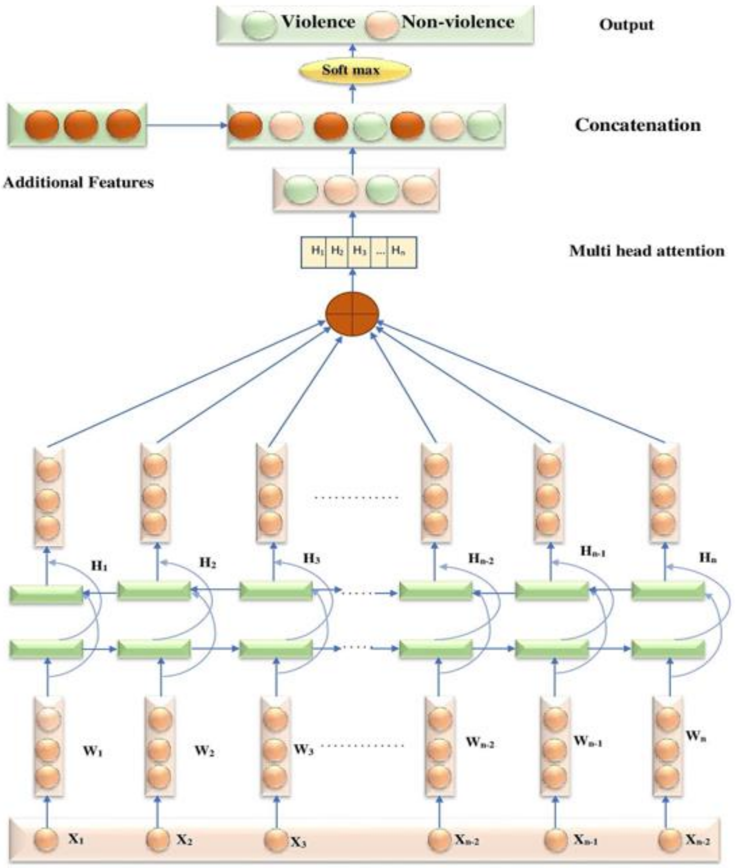


Figure 3. Architecture of Flink framework

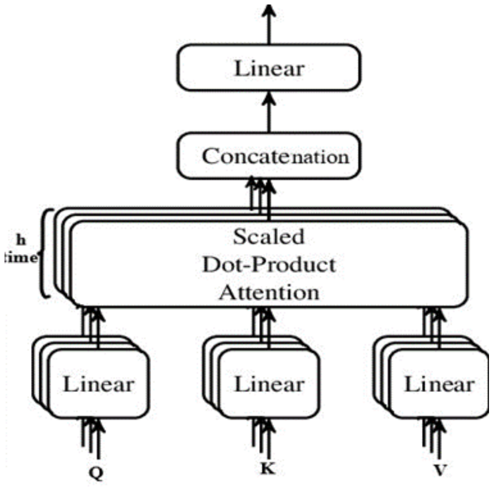


Figure 4. Structure of multihead attention

For violence detection, the resultant vector is routed to the softmax layer. The following is the outcome of the detection:

$$\widehat{z} = softmax(Wz_{gap} + a) \tag{18}$$

The goal of introducing cross-entropy is to assess the types, that shows the difference between the detected violence types y and the detected non-violence types. Structure of multihead attention is shown in Figure 4.

$$l = -\sum_j Z_j \log \widehat{z_j} \tag{19}$$

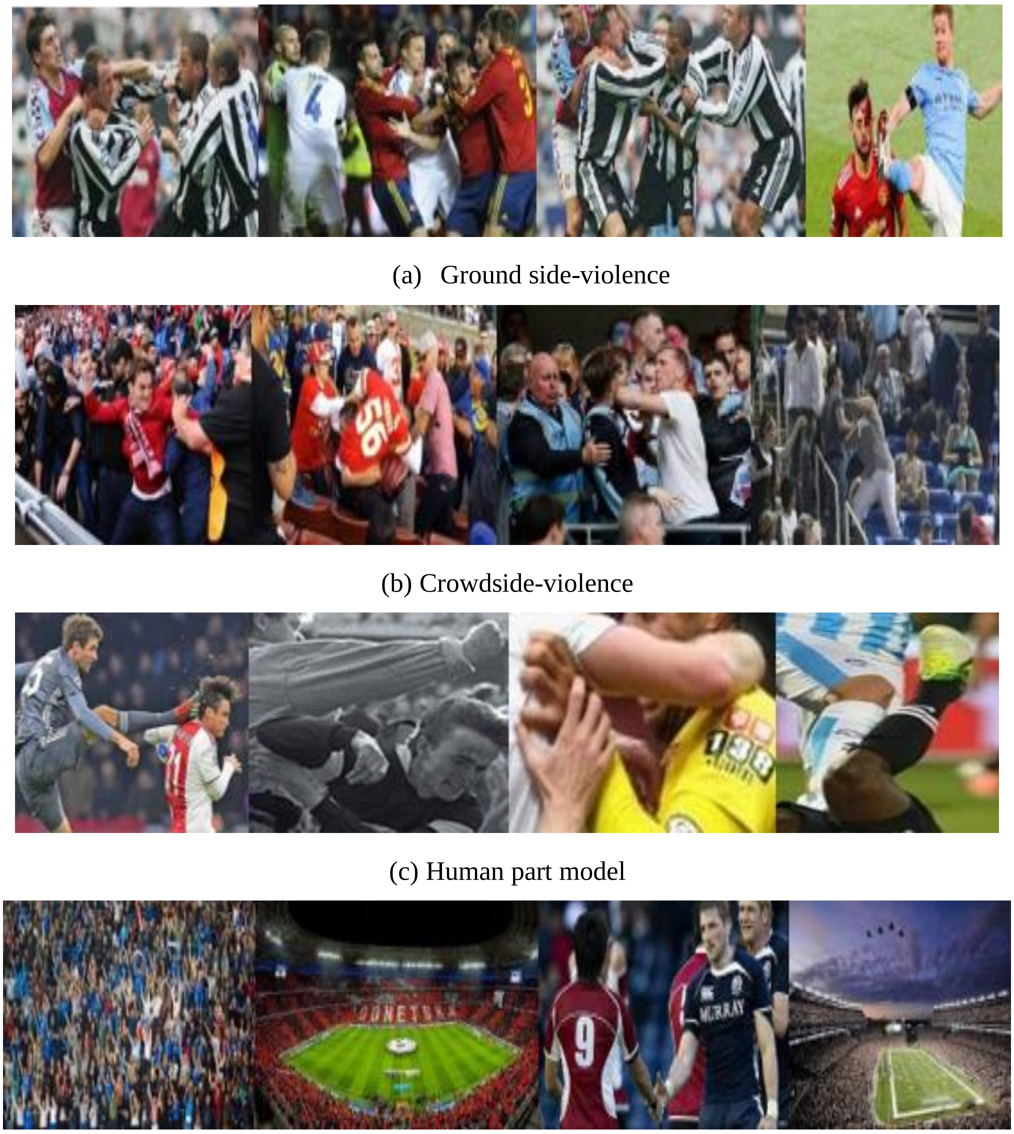
where j is the sentence's index number.

4. RESULTS AND DISCUSSIONS

This part discusses how effective the suggested violence detection system is in identifying violent views in real-time footage. It was designed to detect violence in football stadiums where there is a possibility of violence between players or audience.

Dataset

The dataset is based on the new RWF2000 (Real-World Fighting) dataset from the YouTube website, which contains 2,000 clipped video clips acquired by surveillance cameras from real-world settings. There are 2,000 video clips in the dataset, which are divided into two parts: Half of the videos depict violent acts, while the other half depict non-violent activities. Representation of 4 types of models is shown in Figure 5.



(a) Ground side-violence

(b) Crowdside-violence

(c) Human part model

(d) Non-violence model

Figure 5. representation of 4 types of models

4.1. Performance Evaluation

A comparison is made between the suggested approach and four other methods: Convolutional Neural network Bidirectional LSTM (CNN-BDLSTM), Seperable convolutional LSTM and pre trained mobilenet (SepConvLSTM-M), 3D Convolutional Neural Network with a Support Vector Machines (SVM) classifier(C3D), ResNet50CNN

The network is trained using the retrieved features after extracting them with the HOG function and establishing the MH-BDLSTM parameters. Validation of the violence detection system is performed by calculating precision and recall values. Precision refers to the similarity of two or more measurements to one another. Precision is the word given to a positive predictive value. It's the percentage of retrieved occurrences that are tightly related.

$$Precision(P) = \frac{True\ positive\ rate}{True\ positive\ rate + False\ positive\ rate} \quad (18)$$

Table 1. Comparison of precision for violence detection

Models	**Precision**			
	videos from dataset		**Real-Time**	
	Violence	Non-violence	Violene	Non-violene
CNN-BDLSTM	97.5	96.5	89.2	87.57
SepConvLSTM	97.65	97	89.5	88.1
C3D	97.71	97.54	89.65	88.96
ResNet50 CNN	97.95	97.87	89.72	89.02
Proposed method	98.6	97.96	90.1	90

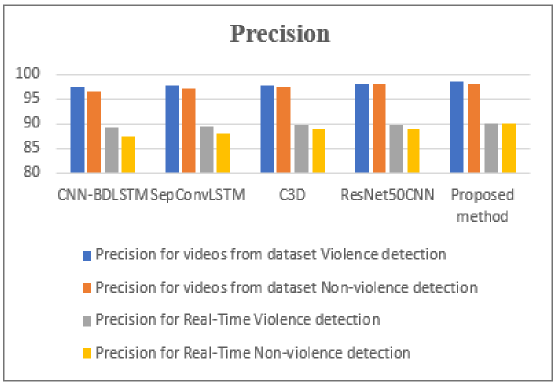


Figure 6. Graphical representation of comparison of precision

Figure 6 shows the precison findings of five different approaches. The proposed violence detection method acquires higher precision results of 90.1% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 89.2 percent, 89.5 percent, 89.65 percent, and 89.72 percent, respectively, as shown in Table 1.

Table 2. Comparison of accuracy for violence detection

Models	Accuracy			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violence	Non-violene
CNN-BDLSTM	95.5	93.5	87.2	86.57
SepConvLSTM	95.1	94	88.5	88.1
C3D	95.51	95.4	89.25	88.96
ResNet50CNN	96.95	96.87	89.52	89.02
Proposed method	98.6	97.96	90.1	90

Figure 7 shows the accuracy findings of five different approaches. The proposed violence detection method acquires higher accuracy results of 90.1% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 87.2 percent, 88.5 percent, 89.25 percent, and 89.52 percent, respectively, as shown in Table 2.

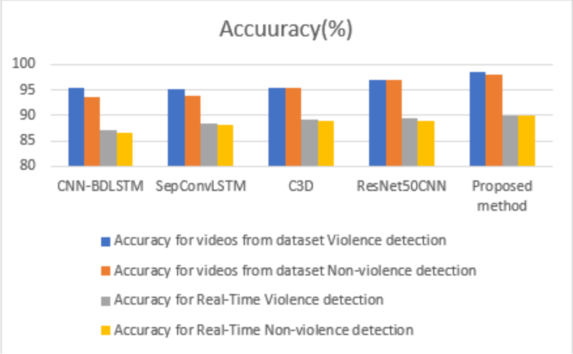


Figure 7. Graphical representation of comparison of Accuracy

Table 3. Comparison of Recall for violence detection

Models	Recall			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violene	Non-violence
CNN-BDLSTM	92.5	91.2	85.2	84.57
SepConvLSTM	92.45	92.4	85.76	85.68
C3D	93.51	93.4	86.25	85.96
ResNet50CNN	94.65	94.17	86.52	86.02
Proposed method	95.42	95.26	87.65	86.90

Recall is the term for sensitivity. The recall is the percentage of relevant examples that might be retrieved that is greater than the entire number of relevant instances. Each accuracy and recall are discussed below based on an understanding and measurement of significance.

$$Recall(R) = \frac{True\ positive\ rate}{True\ positive\ rate+False\ negative\ rate} \qquad (19)$$

Figure 8 shows the recall findings of five different approaches. The proposed violence detection method acquires higher recall results of 87.65% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 85.2 percent, 85.76 percent, 86.25 percent, and 86.52 percent, respectively, as shown in Table 3.

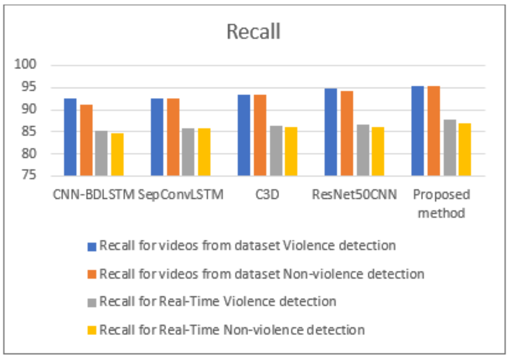


Figure 8. Graphical representation of comparison of Recall

Table 4. Comparison of accuracy for violence detection

Models	Time for execution			
	videos from dataset		Real-Time	
	Violence	Non-violence	Violence	Non-violence
CNN-BDLSTM	25	27	29	30
SepConvLSTM	23	24	26	27
C3D	20	21	23	25
ResNet50CNN	16	18	20	21
Proposed method	11	12	14	15

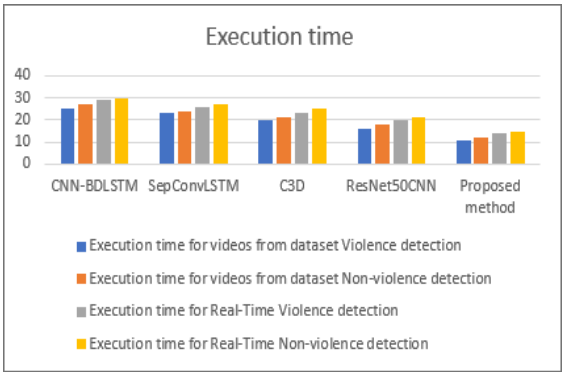


Figure 9. Graphical representation of comparison of execution time

Figure 9 shows the execution time findings of five different approaches. The proposed violence detection method acquires less execution time of 15% for Real time, whereas other methods such as CNN-BDLSTM, SepConvLSTM-M, C3D, and ResNet50CNNprovide precision results of 29 percent, 26 percent, 23 percent, and 20 percent, respectively, as shown in Table 4.

5. CONCLUSION

The rate of violence in football matches has risen dramatically in recent years, whether among players or fans. Security personnel have to be notified in real-time to prevent violence from occurring. Using Flink, the HOG function helps to identify violent behavior by extracting information from video frames. Based on the retrieved features, we can classify the frames into four categories: groundside-violence, crowdside-violence, human part model, and non-violence model. Multihead bidirectional LSTM networks are then trained using the four models to identify violent frames in videos. Compared to a single LSTM, this network arrangement converges faster. When compared with existing systems, the proposed violence detection methodology significantly increases accuracy upto 1.6453%, precision upto 0.646%, recall upto1.959%, and reduces execution time upto 60% than other existing methods. In the future, further research will be conducted to find out if the proposed methodologies can be used to address concerns regarding violence detection.

CONFLICTS OF INTEREST

The author proclaims that there are no conflicting interests to disclose in this study.

FUNDING STATEMENT

Not applicable.

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] 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”, *Plastics*, vol. 19, no. 11, pp. 2472, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] E. Fenil, G. Manogaran, G. N. Vivekananda. Thanjaivadivel, S. Jeeva, and A. Ahilan, “Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM”, *Computer Networks*, vol. 151, pp. 191-200, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Mumtaz, A.B. Sargano, and Z. Habib, “Violence detection in surveillance videos with deep network using transfer learning”, *2^nd European Conference on Electrical Engineering and Computer Science (EECS)*, pp. 558-563, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Rai, A. A. Husain, T. Maity, and R. K. Yadav, “Advance intelligent video surveillance system, (AIVSS): a future aspect”, In Intelligent Video Surveillance. Intech Open, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] B. N. Subudhi, D. K. Rout, and A. Ghosh, “Big data analytics for video surveillance”, *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26129-26162, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data”, *Information Fusion*, vol. 42, pp. 146-157, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] M. M. Najaf Abadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald and E. Muharemagic, “Deep learning applications and challenges in big data analytics”, *Journal of big data*, vol. 2, no.1, pp. 1-21, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[8] B. Jan, H. Farman, M. Khan, M. Imran, I. U. Islam, A. Ahmad, S. Ali and G. Jeon, “Deep learning in big data analytics: a comparative study”, *Computers & Electrical Engineering*, vol.75, pp. 275-287, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Sajjad and S.W. Baik, “CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks”, *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16979-16995, 2021.[CrossRef] [Google Scholar] [Publisher Link]

[10] J. Li, X. Jiang, T. Sun, and K. Xu, “Efficient violence detection using 3d convolutional neural networks”, *16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-8, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] B. M. Peixoto, B. Lavi, Z. Dias, and A. Rocha, “Harnessing high-level concepts, visual, and auditory features for violence detection in videos”, *Journal of Visual Communication and Image Representation*, pp. 103174, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] W. Ullah, A. Ullah, T. Hussain, K. Muhammad, A. A. Heidari, J. Del Ser, S. W. Baik, and V. H. C. De Albuquerque, “Artificial Intelligence of Things-assisted two-stream neural network for anomaly detection in surveillance Big Video Data”, *Future Generation Computer Systems*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] J. Cameron, M. E. Kaye, and E. Scheme, “Dynamic prioritization of surveillance video data in real-time automated detection systems”, *Expert Systems with Applications*, vol. 161, pp.113672, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] A. R. M. Guedes and G. C. Chávez, “Real-Time Violence Detection in Videos Using Dynamic Images”, *In XLVI Latin American Computing Conference (CLEI)*, pp. 503-511, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] P. Wang, P. Wang, and E. Fan, “Violence detection and face recognition based on deep learning”, *Pattern Recognition Letters*, vol. 142, pp. 20-24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] K. Deepak, L. K. P. Vignesh and S. Chandrakala, “Autocorrelation of gradients-based violence detection in surveillance videos”, ICT *Express*, vol. 6, no. 3, pp. 155-159, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Z. Islam, M. Rukonuzzaman, R. Ahmed, M. Kabir and, M. Farazi, “Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM”. arXiv preprint arXiv:2102.10590, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] A. Mumtaz, A. B. Sargano, and Z. Habib, “Violence detection in surveillance videos with deep network using transfer learning”, *2nd European Conference on Electrical Engineering and Computer Science (EECS)*, 2018, pp. 558-563, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] A. M. R. Abdali, and R. F. Al-Tuma, “Robust real-time violence detection in video using cnn and lstm”, 2nd Scientific Conference of Computer Sciences (SCCS) 2019, pp. 104-108, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] T. Z. Ehsan, and S. M. Mohtavipour, “Vi-Net: A Deep Violent Flow Network for Violence Detection in Video Sequences”, *11th International Conference on Information and Knowledge Technology (IKT*), 2020, pp. 88-92, 2020. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

M. Dhipa working as Associate Professor in the Department of Biomedical Engineering at Nandha Engineering College, Erode, Tamil Nadu, India. She completed B.E (EIE) in Easwari Engineering College, Madras University, Chennai in 2004 and M.E (Applied Electronics) in K. S. R. College of Technology, Anna University, Chennai in 2006. She pursued Ph.D. under Anna University, Chennai, Tamil Nadu. She is having 16 years of teaching experience in various institutions. She has published about 12 papers in various International Journals. Her area of interest includes wireless networks.

D. Anitha working as Associate Professor in the Department of Information Technology at Muthayammal Engineering College, Namakkal, Tamil Nadu, India.

Arrived: 18.11.2023

Accepted: 20.12.2023