ISSN 2723-6471

HU Variance Moment Optimizes Keyframe Selection Based on Deep Learning for Violence Detection

Sukmawati Anggraeni Putri[1,*], Pulung Nurtantio Andono[2], Purwanto[3], Moch Arief Soeleman[4]

[1,2,3,4]*Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang 50131, Indonesia*

[1]*Faculty of Technology Information, Universitas Nusa Mandiri, Jakarta 13620, Indonesia*

(Received: November 21, 2024; Revised: December 11, 2024; Accepted: January 21, 2025; Available online: March 4, 2025)

Abstract

Violence in public spaces poses a serious threat to individuals and society. Manual monitoring and violence detection require much time and human resources, ultimately hindering detection accuracy and speed. Therefore, an automated method is needed to detect violence to ensure fast and efficient action. Along with technological advances, violence detection research has adopted various methods and models, including deep learning, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). In this study, the classification process for detecting violence and non-violence uses the VGG19 model, one of the CNN models that has good performance with limited computing. In addition, the Long Short-Term Memory (LSTM) model is the best RNN model for processing temporal data in videos. However, this performance will decrease with noise and irrelevant data in the classification process. Therefore, to optimize deep learning performance, this study in the pre-processing phase selects keyframes in frame extraction using the Hu Variance Moment Technique. This method calculates each frame's Hu and Variance Moment values and selects keyframes based on high Hu values. Next, we use Adaptive Moment Estimation (Adam) to optimize the gradient of the selected keyframes. This study produces a Hu19LSTM model tested on three datasets: hockey fight, crowd, and AIRTLab. The proposed Hu19LSTM model produces an accuracy of 97% on the Hockey Fight dataset, 97% on the Crowd dataset, and 95% on the AIRTLab dataset. These results indicate that the Hu19LSTM model can increase its accuracy on the hockey fight and Crowd dataset by 97%.

Keywords: Violence Detection, Hu Variance Moment Technique, VGG19, Long Short-Term Memory, Adaptive Moment Estimation

1. Introduction

Violence in public spaces threatens personal and social security. Individual greed, frustration, wrath, and economic issues are among the factors contributing to increased public violence. Smart cities employ numerous CCTVs for various purposes, including traffic management and mitigating increased violence in public spaces such as schools, hospitals, and congested centers. According to research by EMP Pusiknas Bareskrim Polri, Indonesia experienced 345,284 crimes between January and October 2024, with violent incidents accounting for 37,712.

The utilisation of artificial intelligence technology in CCTV and computer vision has experienced a substantial increase, particularly in the detection of violent offences. Automated surveillance systems are essential to respond efficiently to acts of violence by identifying anomalies in video data [1]. Therefore, manual surveillance systems have limitations, such as the limited ability of CCTV operators to monitor all incidents recorded by CCTV in public spaces directly. This is where artificial intelligence can be of help. Furthermore, the restricted capacity of operators to concentrate is a disadvantage of manual surveillance. The ability to focus at a CCTV monitoring centre is at its peak for approximately 20 minutes; then it begins to deteriorate [2].

Since "anomaly" is frequently ambiguous and poorly defined, identifying anomalies in the video can be challenging [3]. Violence detection is one of the important parts and components of anomaly detection in videos [4]. As concerns about public safety increase, the use of video surveillance for individual safety is becoming increasingly important,

***Corresponding author: Sukmawati Anggraeni Putri (p41201900017@mhs.dinus.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i2.648

and quickly revealing violent incidents can reduce the risk substantially. The primary aim of a violence detection system is to recognise atypical behaviours classified as violent [5].

Violence An incident's behaviour that deviates from expectations or what is deemed normal is referred to as violence. This type of behaviour includes actions such as hitting, kicking, or dragging another person [6]. Indicators of violent incidents include irregular or abrupt movements, toppling objects, and unusual placement of objects. Establishing automatic, real-time detection of violent incidents is essential to avert potential catastrophes. Continuous and exhaustive monitoring of hostile activities through cameras is difficult for humans, mainly due to their rapidity and repetition [7].

Many researchers have studied various methods to improve the performance of violence detection [8]. Researchers have developed several approaches to detect violent acts in videos over the last decade. We need to classify, describe, and summarize these approaches. As of January 2016, most research on violence detection relied on traditional methods based on previous research results. Earlier studies focused on manually extracting spatial and temporal information from videos. Of all the algorithms used to detect video violence, approximately 24% of previous studies used SVM methods. Between 2015 and 2018, about 25% of all approaches were conventional. Algorithms such as k-nearest neighbours, adaptive boosting, random forest, and k-means are the four most-used machine-learning approaches for violence detection [9]. Devices in computing hardware performance have increased the popularity of deep learning approaches in video-based violence detection. Approximately 43% of all algorithms use deep learning to detect violence, with convolutional neural networks being the most used approach to solve the problem [10].

Deep learning approaches in video analytics for violence detection employ sophisticated but successful spatial and temporal data interpretation and understanding algorithms. Convolutional neural networks (CNNs), which are neural networks that have shown significant success in managing spatial data, are one of the main methods [11], [12]. According to Sharma [13], CNN consists of convolutional layers that apply filters to identify features in pictures or video frames, such as edges, textures, and patterns. CNNs take on the hierarchical character of images motivated by the anatomical arrangement of the human visual cortex [14]. CNN usually has several layers: convolutional, pooling, and fully connected. These layers recognise images, extract features, and minimize spatial dimensions. CNNs have become popular in violence detection due to their ability to recognise intricate patterns and spatial connections [15]. Researchers have created several CNN techniques in the last ten years, such as 3DCNN 3DCNN [16], MobileNet [17], VGG16 [18], and RestNet [11].

Meanwhile, a Recurrent Neural Network (RNN) is used to analyze temporal data [1]. Sudhakaran [19] explains that RNN handles sequential data, such as video, by storing information from previous frames to help analyze subsequent frames. Long Short-Term Memory (LSTM) is used in violence detection systems to analyze sequential video data, allowing the emulation of information about the dynamics associated with different actions. The ability of this network to maintain long-term relationships while managing irrelevant information is critical in distinguishing actual violent incidents from other potentially similar behaviours. LSTM can also overcome the gradient decay problem, making capturing long-term relationships in sequential data practical. In the training process, learning rate optimization methods such as Adaptive Moment Estimation (Adam) optimize the gradient of the image before it is processed in LSTM.

The goal of this research is to develop a model capable of identifying between violent and non-violent circumstances, as well as between different types of violence, including fights and assaults. Surveillance cameras often use video to detect violence. However, the video also contains irrelevant information, such as detecting violence, which requires sending a limited number of frames. An approach to overcome this problem is to extract frames from the video. In previous research, frame extraction and feature extraction process processes performed simultaneously using a Convolutional Neural Network (CNN) showed decreased accuracy [20]. This decrease is due to the possibility of high redundancy in the frame extraction process. We are applying the Hu Variance Moment technique [21] in one solution to reduce. The Hu Variance Moment technique provides a Hu value for each frame generated in the frame extraction process. The calculation of the Hu moment is relatively efficient and does not require significant computing resources to be applied to each frame. This can overcome the problem of frame extraction, which takes a long time to analyze each frame individually. In addition, Hu Moment is quite informative in distinguishing the shape and structure of the

 ISSN 2723-6471

frame and is quite good at handling redundancies. Selecting keyframes based on the highest Hu value is effective and efficient. Keyframes are chosen as the primary input in the learning and classification process to improve the ability of deep learning [20].

Video data processing involves spatial and temporal data that requires a model to manage both data types effectively. In this study, the Long Short-Term Memory (LSTM) model, a Recurrent Neural Network (RNN) variant, is used to manage the temporal relationship between video frames. Keyframes generated from the frame extraction process from videos often show rapid feature changes between keyframes. LSTM is designed to handle this temporal dependency so that the model can consider feature changes over time, an important aspect of video analysis. Next, the features generated from LSTM are processed using the final fully connected layer of the VGG-19 method for final classification to detect violent acts in videos. VGG-19, consisting of 19 layers (16 convolutional layers and three fully connected layers), serves as a spatial feature extraction network, especially visual from video framing. VGG-19 eliminates the need for manual feature extraction, which usually requires in-depth knowledge to select the best combination of features in classification tasks. During training, a learning rate optimization method such as Adaptive Moment Estimation (Adam) is used to update the model weights based on a chosen loss function, such as cross-entropy loss [22]. The model is trained on a pre-labelled dataset to maximize prediction accuracy. The proposed Hu19LSTM model is an improvement that integrates the deep learning algorithms of VGG19 and LSTM with Hu Moment to enhance performance through optimised frame extraction and keyframe selection. This model achieves an accuracy of 97%, a better result than several other models, such as RestNet, VGG-16, and MobileNet with LSTM. Three distinct datasets are used to assess the suggested approach: the 200 video clips in the "Crowd" dataset [23], the "Hockey Fights" dataset with 1000 videos [16], and the "AIRTLab" dataset with 246 video clips [24].

2. The Proposed Method

2.1. Data Pre-Processing

This study performs pre-processing focusing on frame selection and keyframes using optical flow and Hu Variance Moment, which ensures that violent moments are always well represented. This process allows the Model to learn well even without imbalance handling techniques. However, in the three datasets used, the frame extraction process has the potential to produce an imbalanced distribution of violent and non-violent frames. The frames of the captured visual content are extracted and scaled to a dimension of 200×200 pixels (x × y). Numpy3 arrays are used, where each row identifies a sequence or pattern in the film that forms the training data. The sequences can be physical movements or actions, such as reaching out to punch someone, shaking hands, etc. Sequences can only be recorded with a minimum of two frames. This study uses ten consecutive frames, denoted by “n”, to extract temporal variables related to time. The variable N represents the total number of samples in the dataset, calculated by dividing the total frames by the number of frames that make up the sequence. In a simple implementation, NumPy can randomly select the value -1. Thus, the data structure consists of 10 consecutive frames, each with a corresponding class label. The training data dimension is (-1, N, x, y, c), where -1 indicates an undefined value, N is the number of samples, x and y are spatial dimensions, and c refers to the number of channels in each frame.

3.1.1. Frame Extraction

Video surveillance systems detect violence by extracting video into frames using Fixed Interval Sampling, Scene Change Detection, and Optical Flow. This study uses Optical Flow because it effectively captures significant motion, especially in videos with lots of activity. Optical flow is calculated using the intensity changes in two consecutive frames, assuming that the object moves along optical flow lines and the pixel intensity remains constant over time. The basic formula for Optical Flow is as follows:

$$\Delta \mathrm{I} = \mathrm{I}(\mathrm{x}+\mathrm{u}, \mathrm{y}+\mathrm{v}, \mathrm{t}+\Delta \mathrm{t}) - \mathrm{I}(\mathrm{x}, \mathrm{y}, \mathrm{t}) = 0 \quad (1)$$

$I(x, y, t)$ is the image intensity at the position (x, y) at time t. Meanwhile, $I(x+u, y+v, t+\Delta t)$ shows the intensity at the shifted position $x+u, y+v$ in the next frame of time $t+\Delta t$. In this case, u and v are the optical flow components that describe the horizontal and vertical motion of the object that we want to calculate. Meanwhile, ΔI refers to the change in intensity between the two positions.

 ISSN 2723-6471

3.1.2. Keyframe Selection

Keyframe selection is performed after frame extraction using frame difference, entropy, or moment variance methods. The Hu19LSTM Model utilizes the Hu (Histogram of Uniformity) moment variance [21], [20], which is robust to noise, invariant to transformation, and computationally efficient (reducing the risk of overfitting by avoiding the use of redundant or repetitive data while ensuring the Model receives high-quality data that is varied enough to overcome underfitting). Compared to other methods, the Hu moment variance is more suitable for visual structure analysis, as it generates seven values of normalised central moments that help detect patterns and shapes in images. The Hu moment variance remains invariant to translation, scaling, and rotation, making it a practical image and object recognition analysis. In addition, the first absolute orthogonal invariant of the Hu moment variance becomes a fundamental feature descriptor for motion shapes in density distribution analysis. The orthogonal invariant of the first absolute Hu variance moments for the density distribution function is defined as follows:

$$\theta = (\eta 20 + \eta 20) \tag{2}$$

$$\eta pq = \frac{\mu pq}{(\phi 00^{p})}, \rho \frac{(p+q)}{2+1} \tag{3}$$

$$\mu pq = \sum_{u=1}^{u} \sum_{v=1}^{v} u - \ddot{u}^{p} v - \tilde{v}^{q} \, g(u,v), p.q = 0, 1, 2, \ldots \tag{4}$$

$$\varphi pq = \sum_{u=1}^{u} \sum_{v=1}^{v} u^{p} v^{q} \, g(u,v), p.q = 0, 1, 2, \ldots \tag{5}$$

Object orientation (θ) is calculated as the sum of the moment contributions on the horizontal (η20) and vertical (η02) axes. The normalised variance moment (ηpq) is obtained by dividing the central moment (μpq) by the normalised zero-order central moment (ϕ00), adjusted based on the orders of p and q. The central moment (μpq) calculates the intensity distribution around the image centre of mass by summing the pixel contributions whose intensity is multiplied by the difference between the pixel coordinates and the centre of mass. The corrected moment (φpq) calculates the pixel contributions based on the coordinates without considering the centre of mass. All these moments describe the shape and distribution of objects in the image. The K-means technique clusters the data based on the extracted attributes to determine the final keyframe set. The system determines the cluster distance as an initial step and then uses the Silhouette Coefficient (SC) index to confirm the ideal number of clusters (k value). The k value determines keyframes, where each cluster represents one frame in the video.

3.1.3. Feature Extraction

Convolutional Neural Networks (CNN) effectively extract spatial features and outperform manual feature extraction techniques, which often struggle to select the optimal combination of features for classification. With automatic feature extraction, CNN improves classification accuracy [25]. One of the CNN architectures, the VGG19 Model, was chosen over ResNet or MobileNet because this study focuses on obtaining better feature representations from the dataset while maintaining stable and straightforward computational requirements. The VGG19 Model, extracts local features using a 3×3 small filter through nine convolutional layers. This process produces features in the final pooling layer from important frames of the video footage. Feature optimization is enhanced using the Adaptive Moment Estimation (Adam) algorithm, which combines the advantages of the AdaGrad and RMSProp methods [22]. ADAM is better in adaptability, efficiency on big data, and stability in handling noise and sparsity. Compared with SGD, Momentum, Adagrad, and RMSProp, ADAM offers the best balance between convergence speed, adaptivity, and stability, thus improving machine learning performance. The basic formula of Adam is:

Moment Update Calculation

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{6}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{7}$$

Bias Correction Calculation:

$$m_t = \frac{m_t}{1 - \beta_1^t} \tag{8}$$

 ISSN 2723-6471

$$v_t = \frac{v_t}{1-\beta_2^t} \tag{9}$$

Calculation of Keyframe parameter updates:

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \epsilon}} \tag{10}$$

At step t, gt represents the gradient function loss. The moment calculation process includes two components: mt (first moment) as the gradient moving average and vt (second moment) as the squared gradient moving average. In its classification, the ADAM method calculates the gradient of the loss function to assess the quality of keyframes based on the Hu Variance Moments. By adjusting the learning rate using mt and vt, ADAM improves the efficiency and stability of optimisation, accelerates convergence, and ensures that the selected keyframes are the most representative and informative during training.

2.2. Deep Learning Classification

The proposed Hu19LSTM model uses the LSTM network to develop a deep learning model capable of performing classification based on features extracted from the VGG19 convolutional base. This model enhances the capabilities of LSTM [26] by integrating convolutional structures on the transitions between states and from input to state. This approach, known as ConvLSTM, more effectively combines spatial and temporal information. ConvLSTM is designed to extract spatial and temporal correlations better than traditional LSTM models, making it superior in predicting future violent events. The proposed ConvLSTM model consists of a single ConvLSTM layer with 128 filters and a 5 x 5 kernel designed to optimize prediction accuracy. The following equations (10) to (14) describe the LSTM function with input X_t, cell output C_t, hidden state H_t, and gates i_t, f_t, o_t, as well as the convolution "$*$" and the Hadamard "$\odot$".

$$\mathrm{i_t} = \sigma(\mathrm{W_{xi}X_t} + \mathrm{W_{hi}H_{t-1}} + \mathrm{b_i}) \tag{11}$$

$$\mathrm{f_t} = \sigma(\mathrm{W_{xf}X_t} + \mathrm{W_{hf}H_{t-1}} + \mathrm{b_f}) \tag{12}$$

$$\mathrm{o_t} = \sigma(\mathrm{W_{xo}X_t} + \mathrm{W_{ho}H_{t-1}} + \mathrm{b_o}) \tag{13}$$

$$\mathrm{C_t} = \mathrm{f_t} \odot \mathrm{C_{t-1}} + \mathrm{i_t} \odot \tanh(\mathrm{W_{xc}X_t} + \mathrm{W_{hc}H_{t-1}} + \mathrm{b_c}) \tag{14}$$

$$\mathrm{H_t} = \mathrm{o_t} \odot \tanh(\mathrm{C_t}) \tag{15}$$

LSTM uses three main gates to manage information: the input gate (i_t), the forgetting gate (f_t) and the output gate (o_t). The input gate (i_t) controls how much new information from the input (X_t) and the previous hidden state (H_t) enters the cell state (C_t), with the sigmoid function (σ) keeping the value between 0 and 1. The forgetting gate (f_t) determines the information to be forgotten from the previous cell state (C_{t-1}). The cell state (C_t) is updated by combining the forgotten part of (C_{t-1}) and the new information from the input (X_t) and the hidden state (H_{t-1}). The forgotten information is $f_t \odot C_{t-1}$, while the new information added is $i_t \odot \tanh(W_{xc}X_t + W_{hc}H_{t-1} + b_c)$. The output gate ($o_t$) controls the information from the cell state (C_t) that is used to generate the hidden state (H_t), where H_t is obtained from $o_t \odot \tanh C_t$. This allows the LSTM to control the information used in the next step.

3. Methodology

This section discusses in detail the experimental design used in this study. A proper experimental design is important to ensure the validity of the data obtained and to answer the research questions accurately and systematically.

3.1. Data

The utility of the Hu19LSTM architecture is tested by applying it to three standard datasets for the detection of violence and non-violence actions. First, the Hockey Fights dataset [18], is video data from hockey games. Second, the Crowd dataset [27] is video data from various films that focus on action scenes. Third, the AIRTLab dataset [24] is video data from a simulation room used to train and test a violence detection system.

ISSN 2723-6471

3.1.1. Hockey Fight Dataset

Researchers used the “Hockey Fights” dataset, which consists of hockey game videos that include 500 violence and 500 non-violence videos, with an average duration of one second. All videos have similar contexts and discuss the same issues [18]. Figure 1 shows examples of violent and non-violent videos from the dataset.

Figure 1. The Examples of Violence (first row) and Non-violence (second row) Frames from the Hockey Fight Dataset

3.1.2. Crowd Dataset

The Crowd dataset consists of video clips from a variety of films that explicitly focus on action scenes. In contrast, the non-violent clips in this dataset come from clips whose primary goal is action recognition. This dataset consists of 100 clips with violent content and 100 clips with non-violent content, each with an average duration of 1 second. In contrast to the Hockey Fights dataset, these video clips feature a variety of backgrounds and themes [27]. Figure 2 shows examples of violence and non-violence clips from the Crowd dataset.

Figure 2. The Examples of Violence (first row) and non-violence (second row) Frames from the Crowd Dataset

3.1.3. AIRTLab Dataset

A collection of 350 short videos categorised as either "violent" or "non-violent". These videos were created to train and test a system that can detect violence in videos. Each video is 1080p and recorded from two different camera angles to provide a wider range of perspectives. Violent actions include hitting and kicking, while non-violent actions include hugging and shaking hands, which can often cause the system to misidentify violence. This dataset helps the system to distinguish more accurately between real violence and similar but harmless gestures. Figure 3 shows examples of violent and non-violent clips from the AIRTLab dataset [24].

Figure 3. The Examples of Violence (first row) and Non-Violence (second row) Frames from the AIRTLab Dataset

3.2. Environment and Setting

This section outlines the environment and settings used in this study. Selecting the right environment and settings is crucial to ensure the data generated remains consistent, valid, and reliable throughout the study. The proposed Hu19LSTM model, as shown in figure 4, must go through three phases to detect acts of violence: pre-processing, training, and testing.

 ISSN 2723-6471

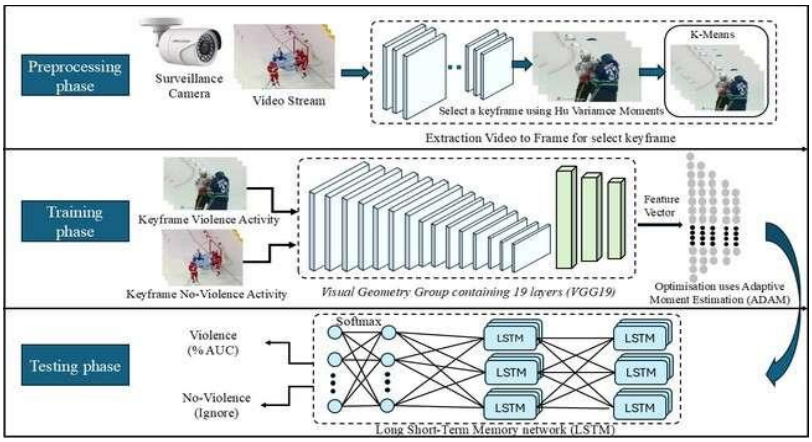


Figure 4. The Proposed Hu19LSTM Framework for Violence Detection

In figure 4, the preprocessing stage begins (first phase) by extracting the video from the surveillance camera into 200 x 200-pixel frames to obtain spatial and temporal information. The system uses optical flow to analyze the motion patterns between frames by calculating the motion vectors between pixels based on the magnitude and direction of the motion. Optical flow identifies frames with significant activity so that only relevant frames are selected to represent the video efficiently. After the frames are extracted, the system applies the Hu Variance Moments method to measure the variation of features in each frame, select keyframes that contain essential information, and filter out frames that include noise, such as insignificant random patterns. Frames with Hu Variance Moments values are grouped using the K-Means algorithm to ensure that keyframes are optimally selected to represent violent and non-violent activities. Furthermore, the frame with the highest Hu variance moment value is chosen as the keyframe, which aims to reduce redundancy and noise by eliminating irrelevant frames. This process produces a set of keyframes for Model training in the next stage.

In the training phase (second phase), keyframes generated from the previous stage are grouped into two main categories: violent and non-violent activities. The system utilizes the VGG19 Model, a convolutional neural network of 19 layers, to extract visual features from each keyframe. This Model analyses visual elements such as shape, texture, and patterns in the image. Each layer in VGG19 is designed to detect features at different levels, starting from simple features such as edges and lines in the early layers to complex patterns such as specific objects or scenes in deeper layers. The result of this process is a feature vector, a numeric representation that reflects the unique visual characteristics of each keyframe. After the feature extraction, the system uses the Adaptive Moment Estimation (ADAM) optimization algorithm to refine the Model parameters. ADAM works by adjusting the weights in the neural network based on the error gradient generated during training, making the learning process faster and more stable. ADAM utilizes a combination of gradient momentum and gradient mean square to optimize parameter updates, accelerating Model convergence and improving accuracy. At the end of this phase, the system produces a trained Model that can recognize visual patterns well enough to classify violent and non-violent activities accurately.

In the testing phase (thrid phase), the model feeds keyframes from the second-phase process into the training Model to analyze the temporal relationship between keyframes using LSTM (Long Short-Term Memory). LSTM processes each frame sequentially, retaining the context of previous frames with its internal memory and ignoring irrelevant information. This process allows LSTM to capture temporal patterns, such as significant changes in the sequence of keyframes and variations in Hu Moments. This helps identify essential events and enhance the clarity of action in the video. After the temporal analysis, two fully connected layers with 512 and 2 neurons are used, followed by the Softmax activation function to predict the video category into two classes: “Violence” and “No-Violence”. The prediction results are measured by the percentage of AUC (Area Under Curve) to ensure the accuracy of the classification. The model optimizes the learning process using the sparse categorical cross-entropy loss function, designed for binary classification with labels’ 0’ for non-violence and ‘1’ for violence. This approach utilizes data in frames or feature sequences containing temporal patterns so the Model can effectively learn the differences between categories. Each iteration is performed with a batch size of five samples, and the data is divided into 80% for training and 20% for

 ISSN 2723-6471

testing. Training is carried out for 200 epochs to ensure an optimal tracking Model, resulting in accurate and efficient video classification.

3.3. Evaluation

This study uses accuracy metrics, and area under the curve (AUC) [27] to compare the proposed method with current best practices, which is described in the following equation:

$$\text{Accuracy} = \frac{\text{True Positive (TP)+ True Negative (TN)}}{\text{True Positive (TP)+ True Negative (TN)+ False Positive (FP)+ False Negative (FN)}} \quad (16)$$

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)+ False Positive (FP)}} \quad (17)$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)+ False Negative (FN)}} \quad (18)$$

The model classifies data as positive and true positive, resulting in a True Positive (TP) value. Conversely, False Positive (FP) occurs when the model classifies data as positive but negative. True Negative (TN) shows the number of correct predictions when the model classifies data as negative, and the result is negative. False Negative (FN) occurs when the model classifies data as negative when it is positive. These four metrics are very important for measuring and describing the prediction model's level of accuracy. A good model should produce a high TP value and low FP and FN values, thus indicating its ability to minimize prediction errors. In addition, the Area Under the Curve (AUC) is used to evaluate the model's ability to distinguish between true positive and negative data.

4. Results and Discussion

This section discusses the effectiveness of the proposed Hu19LSTM Model on three violence datasets. The evaluation process involves measuring accuracy and AUC to compare the proposed model's performance with models from previous studies. This study develops the Hu19LSTM Model to optimise deep learning models, namely the Convolutional Neural Network (CNN) VGG19 approach and the RNN LSTM approach. This model processes video data to detect violence by extracting frames and selecting keyframes. The extracted video, which consists of a collection of individual frames, is then selected as a keyframe using the Hu Variance Moment technique to determine the most relevant and significant keyframes and reduce noise in the learning process. The Hu Variance Moment technique calculates two main parameters: the Histogram of Uniformity (HU) and Variance Moment (VM). The HU value describes the distribution of pixel intensity in the image, while the VM value represents the variation in pixel intensity. A high HU value indicates good contrast in the frame selection process, while a low VM value indicates a more homogeneous texture. In addition, changes in HU or VM values can reveal significant differences between frames. The standard HU values of 0.5 - 0.8 is a good contrast. Meanwhile, the VM value of 0.2 - 0.4 is good homogeneity. The basic criteria can be determined in keyframe selection according to the general standard limits. The keyframe selection process for the Hockey Fight Dataset was done by applying Hu Variance Moment to 500 videos covering 50,000 frames. Keyframe selection criteria are HU $\geq$ 0.7 and VM $\leq$ 0.3. The calculation results are presented in table 1.

Table 1. The Results of HU Variance Moment Calculation on Hockey Fight Dataset

Frame	HU	VM	Result
1	0.72	0.28	Keyframe
2	0.41	0.59	Not criteria
3	0.85	0.15	Keyframe
…	…	…	…
50,000	0.67	0.33	Keyframe

Table 1 shows that frames with high HU and low VM values represent intense fighting scenes, while frames with low HU and high VM values depict transitional or less relevant scenes. The keyframe selection criteria successfully identified important moments in the fight. Based on the analysis of table 1, the number of selected keyframes was

ISSN 2723-6471

15,000 (30%) with an average HU value of 0.75, an average VM value of 0.25, a standard deviation of HU of 0.12, and a standard deviation of VM of 0.15. The keyframe selection process uses Hu Variance Moment on the Crowd dataset of 100 videos with 10,000 frames. Keyframe selection uses the criteria HU ≥ 0.7 and VM ≤ 0.3. Table 2 presents the results of the calculation.

Table 2. Results of HU Variance Moment Calculation on Crowd Dataset

Frame	HU	VM	Result
1	0,81	0,19	Keyframe
2	0,42	0,58	Not criteria
3	0,91	0,09	Keyframe
…	…	…	…
10,000	0,75	0,25	Keyframe

Based on the results presented in table 2, frames with high HU and low VM values represent dense crowds, while frames with low HU and high VM values indicate transitions or unimportant scenes. The keyframe selection criteria have proven effective in identifying key moments in the crowd. From the analysis of table 2, the number of keyframes is 3,000 (30%), with an average HU value of 0.72, an average VM value of 0.28, a standard deviation of HU of 0.11, and a standard deviation of VM of 0.14.

Hu Variance Moment calculation was performed on the AIRTLab Dataset, consisting of 350 videos with 35,000 frames. The key frame selection process uses the criteria of HU ≥ 0.7 and VM ≤ 0.3. The calculation results are presented in table 3.

Table 3. The Result of HU Variance Moment on AIRTLab Dataset

Frame	HU	VM	Result
1	0.81	0.19	Keyframe
2	0.41	0.59	Not the criteria
3	0.91	0.09	Keyframe
…	…	…	…
35,000	0.75	0.25	Keyframe

Based on the results presented in table 3, frames with high HU and low VM values represent intense human activity. In contrast, frames with low HU and high VM values indicate transitional or less relevant scenes. The keyframe selection criteria proved effective in identifying key moments. From the analysis of table 3, the number of keyframes was 10,500 (30%), with an average HU value of 0.74, an average VM value of 0.26, a standard deviation of HU of 0.12, and a standard deviation of VM of 0.15.

Figure 6 shows the keyframe selection process. The original frames in figure 5 are processed to obtain the Hu Variance moment value of each original frame. The selected keyframes are then used in the feature selection and learning process using the model. The deep learning developed is VGG19 one of the CNN methods with low computation and LSTM the best RNN method.

Figure 5. The original Frame before Keyframe Selection using the Hu Variance Moment Technique

ISSN 2723-6471

Figure 6. The sample keyframes that have been selected using the Hu Variance Moment Technique

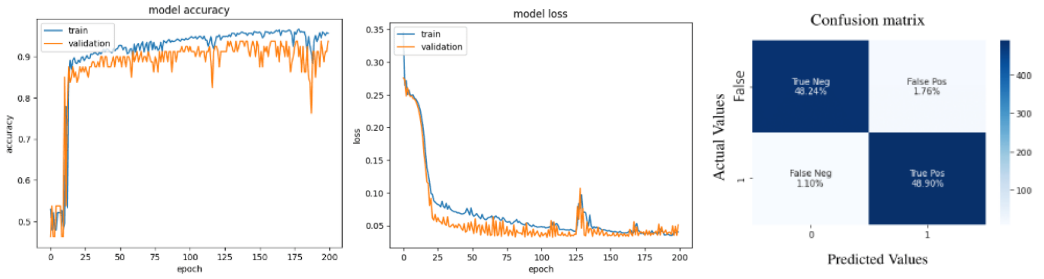
This study tests the proposed Hu19LSTM Model on three benchmark datasets: Hockey Fight, Crowd, and AIRTLab. The dataset is split in an 80:20 ratio, with 80% training, and 20% testing data. In the pre-processing phase, the selected keyframes are used as the basis for training and testing. In this process, the VGG19 model is used to process the visual images of the keyframes, while LSTM is used to analyse the temporal relationship between keyframes. The test results show that the Hu19LSTM Model achieves an accuracy of 95.50% on the Hockey Fight dataset, 92.40% on the Crowd dataset, and 94.50% on the AIRTLab dataset. The testing process considers various important parameters, such as the number of filters and kernel size, which significantly affect the performance of the Hu19LSTM network. The best model is obtained with 128 filters and a kernel size of 5 x 5.

Table 4 presents the results of testing the model on each dataset with various parameters. The Hu19LSTM model performs optimally in detecting violence, especially on the Hockey Fight dataset with 0.97 accuracy and 0.03 loss, and AIRTLab with 0.95, accuracy and 0.05 loss using a 5 x 5 kernel. Meanwhile, on the Crowd dataset, the model produces 0.97 accuracy and 0.03 loss with a 7 x 7 kernel.

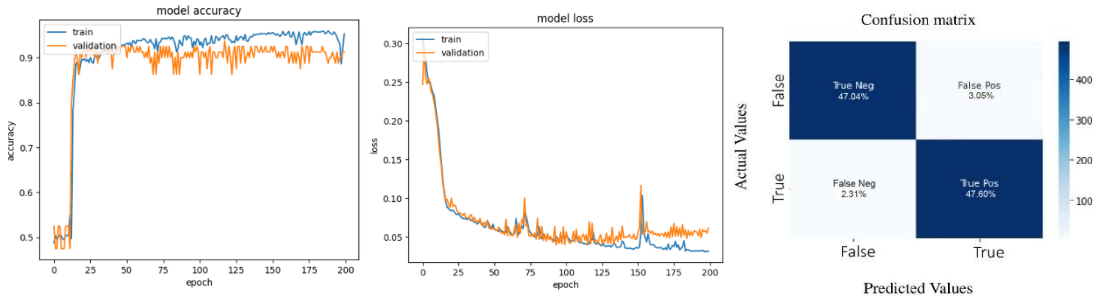
Table 4. The Performance of the HU19LSTM model

Parameter	Hockey Fight		Crowd		AIRTLab	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Filter size = 128, Kernel = 5 x 5	0.97	0.03	0.90	0.08	0.95	0.05
Filter size = 128, Kernel = 3 x 3	0.92	0.08	0.90	0.08	0.91	0.09
Filter size = 128, Kernel = 7 x 7	0.93	0.07	0.97	0.03	0.92	0.08

After the Hu19LSTM Model was tested on three datasets, the accuracy and loss results are shown in figure 7 through the accuracy percentage graph, Model loss, and confusion matrix.

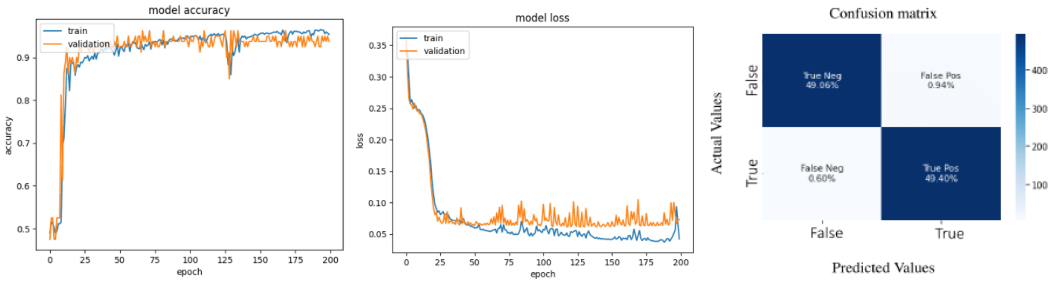


(a) Hokey Fight Dataset



(b) AIRTLab dataset

ISSN 2723-6471



(c) Crowd Dataset

Figure 7. Learning curves of training and validation accuracy and loss and its confusion matrix in Hockey Fight Dataset, AIRTLab Dataset, Crowd Dataset

Figure 7 shows a graph of the accuracy and loss results on the three datasets used: a) Hockey Fight, b) AIRTLab, and c) Crowd. These datasets are public and are used in the training process of the Hu19LSTM model developed in this study. On the Hockey Fight dataset, this model produces an accuracy of 0.97. While on the Crowd dataset, this model obtains an accuracy of 0.97. For the AIRTLab dataset, the accuracy obtained is 0.95. In addition, the low loss value indicates that the Hu19LSTM model has been well-trained and has a low error rate. This study also compares the accuracy results of the proposed Hu19LSTM model with the method produced in previous studies. Table 5 explains that the Hu19LSTM model produces better accuracy than the previous model on the Hockey Fight dataset (0.97) and the Crowd dataset (0.97). However, on the AIRTLab dataset, the Hu19LSTM model is not better than the previous model, with an accuracy value of 0.95.

Table 5. The Comparison of State-Of-The-Art Method and Proposed Method

Dataset	Metric	3D CNN [16]	CNN + LSTM [13]	MobileNetV2 + GRU [17]	Proposed Hu19LSTM	ResNet50 + LSTM [11]
AIRTLab	AUC	0.93	0.91	0.94	0.95	0.94
	Acc	0.93	0.91	0.94	0.95	0.94
	F1	0.93	0.91	0.93	0.94	0.93
	Prec	0.93	0.91	0.93	0.94	0.93
	Recall	0.94	0.92	0.94	0.95	0.94
Crowd	AUC	0.93	0.93	0.95	0.97	0.93
	Acc	0.93	0.93	0.95	0.97	0.93
	F1	0.94	0.94	0.96	0.98	0.94
	Prec	0.94	0.94	0.96	0.98	0.94
	Recall	0.94	0.94	0.96	0.98	0.94
Hockey	AUC	0.93	0.94	0.95	0.97	0.95
	Acc	0.93	0.94	0.95	0.97	0.95
	F1	0.93	0.94	0.95	0.97	0.95
	Prec	0.93	0.94	0.95	0.97	0.95
	Recall	0.94	0.95	0.96	0.98	0.96

**Acc: Accuracy, Prec: Precision*

Table 5 compares the performance of our proposed Hu19LSTM Model with previous studies on the Hockey Fight, Crowd, and AIRTLab datasets. On the Hockey Fight dataset, the Model achieved 97% accuracy with an AUC of 0.97, reflecting a high ability to distinguish between violent and non-violent incidents. The precision of 97% indicates most of the violence predictions were correct (3% false positives), while recall of 98% suggests the Model's effectiveness in detecting violence (2% false negatives). The F1 score of 97% reflects an excellent balance between precision and recall. On the Crowd dataset, the Hu19LSTM Model showed consistent performance with 97% accuracy, AUC 0.97, precision 98% (2% false positives), recall 98% (2% false negatives), and F1-score 98%, reflecting an optimal balance in accurately detecting violent incidents. Meanwhile, on the AIRTLab dataset, the Model achieved 95% accuracy, AUC

 ISSN 2723-6471

0.95, precision 95% (5% false positives), recall 96% (4% false negatives), and F1-score 95%, indicating a continued good performance in distinguishing between violence and non-violence. The proposed keyframe extraction method, using Hu Variance Moments to determine the value of each video frame, identifies the most relevant areas with critical actions in that frame based on the highest Hu Moments value. Experiments show that the keyframe selection approach with the Hu Variance Moments technique in the frame extraction process can recognize scenes in relevant frames. Using more informative, concise, and relevant frames reduces the input dimension, speeds up the training time, and improves the accuracy of video classification with VGG19 and LSTM through effective keyframe selection.

5. Conclusion

This study recommends using a deep learning-based assistant system to detect violent activity in software with limited resources. The developed Hu19LSTM model is an extension of the VGG19 deep learning model, one of the CNN models that provides good performance even with limited computing. LSTM is used to process temporal information in video data and is one of the best RNN models. The study optimizes the classification process by selecting keyframes using the Hu Variance Moment technique from several frames extracted from the video. In addition, we use ADAM optimization in the training process to overcome the interference usually found in frame extraction features. In addition, the training process uses ADAM optimization to overcome interference usually found in frame extraction features. In the classification phase, we compare the proposed model with previously developed models: the CNN LSTM model achieves an accuracy of 0.933, RestNet50-LSTM 0.950, and MobileNetV2-GRU 0.954. Our proposed Hu19LSTM model shows a higher accuracy of 0.955. The results of this study indicate that the Hu19LSTM model, which combines VGG19 and LSTM with Hu Variance Moment optimization, has excellent performance in terms of accuracy, loss, and computing time.

Violence detection in videos remains an active and interesting research topic. Some recommendations for further research include designing new tools to explore or creating large and balanced datasets from multiple video sources to improve violence detection with more categories. This allows for identifying different types of violence rather than just detecting the presence of violence. Future research could also consider using transformer techniques to handle and process complex data sequences, such as interdependent video frames in a time sequence. The simultaneous use of LSTM and Transformer can be done through several methods that utilize each of their advantages. The LSTM can be used as pre-processing for the Transformer by generating a more compact feature representation before the Transformer captures the global relationships. Conversely, the Transformer can serve as pre-processing for the LSTM by capturing global relationships, which the LSTM then utilizes for local temporal information. Other approaches involve hybrid layers, such as the Transformer as an encoder and the LSTM as a decoder, or using the LSTM as a sequence reducer to generate a fixed representation easier for the Transformer to process. The Model can also be trained on private datasets obtained from CCTV footage or simulations with explicit permission. Categorize videos into violent (fights, assaults) and non-violent (peaceful crowds, sports events without incident). Vary the data based on resolution (low to high), duration (3-10 seconds), environment (stadium, street, school, park), and camera angle (CCTV, mobile phone, drone). Engage a team of annotators or use an AI service to assign labels, such as category, type of violent act, duration of the act, and associated objects (weapons or other tools).

The Model that has been built has excellent potential to be applied in live surveillance, such as detecting suspicious activities on CCTV, such as physical violence (fights). However, its application faces challenges such as the need for advanced hardware to process real-time data, the ability to handle big data without losing accuracy, and the risk of misdetection that can lower user confidence. In addition, the Model must be able to adapt to environmental changes, keep sensitive data secure, and require periodic retraining to remain relevant. The Model can be optimally implemented in several surveillance systems by addressing these challenges.

6. Declarations

6.1. Author Contributions

Conceptualization: S.A.P., P.N.A., P., and M.A.S.; Methodology: P.N.A.; Software: S.A.P.; Validation: S.A.P., P.N.A., and P.; Formal Analysis: S.A.P., P.N.A., and P.; Investigation: S.A.P.; Resources: P.N.A.; Data Curation: P.N.A.;

Writing Original Draft Preparation: S.A.P., P.N.A., and P.; Writing Review and Editing: P.N.A., S.A.P., and P.; Visualization: S.A.P. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Feng, Y. Liang, and L. Li, "Anomaly Detection in Videos Using Two-Stream Autoencoder with Post Hoc Interpretability," *Comput. Intell. Neurosci.*, vol. 2021, no. 1, pp. 15, 2021, doi: 10.1155/2021/7367870.

[2] S. A. Velastin, B. A. Boghossian, and M. A. Vicencio-Silva, "A motion-based image processing system for detecting potentially dangerous situations in underground railway stations," *Transp. Res. Part C Emerg. Technol.*, vol. 14, no. 2, pp. 96–113, 2006, doi: 10.1016/j.trc.2006.05.006.

[3] M. Y. Yang, W. Liao, Y. Cao, and B. Rosenhahn, "Video event recognition and anomaly detection by combining gaussian process and hierarchical Dirichlet process models," *Photogramm. Eng. Remote Sensing*, vol. 84, no. 4, pp. 203–214, 2018, doi: 10.14358/PERS.84.4.203.

[4] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik, and M. Y. Lee, "Cover the violence: A novel deep-learning-based approach towards violence-detection in movies," *Appl. Sci.*, vol. 9, no. 22, 2019, doi: 10.3390/APP9224963.

[5] A. Ben Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: a review," *Expert Syst. Appl.*, vol. 4, no. 91, pp. 480–491, 2018, doi: 10.1016/j.eswa.2017.09.029.

[6] Z. Shao, J. Cai, and Z. Wang, "Smart Monitoring Cameras Driven Intelligent Processing to Big Surveillance Video Data," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 105–116, 2017, doi: 10.1109/tbdata.2017.2715815.

[7] Barbara Kitchenham, "Systematic Review in Software Engineering – Where We Are and Where We Should Be Going," in *EAST '12: Proceedings of the 2nd international workshop on Evidential assessment of software technologies*, vol. 2, no. 1, pp. 1 - 2, 2007, doi: 10.1145/2372233.2372235.

[8] M. Biswas *et al.*, "State-of-the-Art Violence Detection Techniques: A review," *Asian J. Res. Comput. Sci.*, vol. 13, no. 1 February, pp. 29–42, 2022, doi: 10.9734/ajrcos/2022/v13i130305.

[9] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight Recognition in Video Using Hough Forests and 2D Convolutional Neural Network," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4787–4797, 2018, doi: 10.1109/TIP.2018.2845742.

[10] P. Contardo, P. Sernani, N. Falcionelli, and A. F. Dragoni, "Deep learning for law enforcement: A survey about three application domains," *CEUR Workshop Proc.*, vol. 2872, no. July, pp. 36–45, 2021. doi: 10.3390/s23041939.

[11] M. Shoaib and N. Sayed, "A Deep Learning Based System for the Detection of Human Violence in Video Data," *Trait. du Signal*, vol. 38, no. 6, pp. 1623–1635, 2021, doi: 10.18280/ts.380606.

[12] U. Usman, F. Yunita, and M. R. Ridha, "Improving Classification Accuracy of Local Coconut Fruits with Image Augmentation and Deep Learning Algorithm Convolutional Neural Networks (CNN)," *Journal Of Applied Data Sciences*, vol. 6, no. 1, pp. 1–19, 2025, doi: 10.47738/jads.v6i1.389.

 ISSN 2723-6471

[13] S. Sharma, B. Sudharsan, S. Naraharisetti, V. Trehan, and K. Jayavel, "A fully integrated violence detection system using CNN and LSTM," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 4, pp. 3374–3380, 2021, doi: 10.11591/ijece.v11i4.pp3374-3380

[14] R. Halder and R. Chatterjee, "CNN-BiLSTM Model for Violence Detection in Smart Surveillance," *SN Comput. Sci.*, vol. 1, no. 4, 2020, doi: 10.1007/s42979-020-00207-x.

[15] M. Ramzan et al.,"A Review on State-of-the-Art Violence Detection Techniques," *IEEE Access*, vol. 7, no. 1, pp. 107560–107575, 2019, doi: 10.1109/ACCESS.2019.2932114.

[16] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, "Violence detection using spatiotemporal features with 3D convolutional neural network," *Sensors*, vol. 19, no. 11, pp. 1–15, 2019, doi: 10.3390/s19112472.

[17] J. Mahmoodi and A. Salajeghe, "A classification method based on optical flow for violence detection," *Expert Syst. Appl.*, vol. 127, no. Aug., pp. 121–127, Aug. 2019, doi: 10.1016/j.eswa.2019.02.032.

[18] S. Mukherjee, R. Saini, P. Kumar, P. P. Roy, D. P. Dogra, and B.-G. Kim, "Fight Detection in Hockey Videos using Deep Network," *J. Multimed. Inf. Syst.*, vol. 4, no. 4, pp. 225–232, 2017, doi: 10.9717/JMIS.2017.4.4.225.

[19] S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," *2017 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2017*, vol. 14, no. 1, pp. 1-7, doi: 10.1109/AVSS.2017.8078468.

[20] Y. Sun, G. Wen, and J. Wang, "Weighted spectral features based on local Hu moments for speech emotion recognition," *Biomed. Signal Process. Control*, vol. 18, no. 80 - 90, pp. 80–90, 2015, doi: 10.1016/j.bspc.2014.10.008.

[21] H. Ming-Kuei, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–188, 1962, doi: 10.1109/TIT.1962.1057692.

[22] S. Letchmunan, U. M. Butt, F. H. Hassan, S. Zia, and A. Baqir, "Detecting video surveillance using VGG19 convolutional neural networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 2, pp. 674–682, 2020, doi: 10.14569/ijacsa.2020.0110285.

[23] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, "Violence Detection in Video Using Computer Vision Techniques," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6855 LNCS, no. PART 2, pp. 332–339, 2011, doi: 10.1007/978-3-642-23678-5_39.

[24] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni, "Deep Learning for Automatic Violence Detection: Tests on the AIRTLab Dataset," *IEEE Access*, vol. 9, no. 1, pp. 160580–160595, 2021, doi: 10.1109/ACCESS.2021.3131315.

[25] Irfanullah, T. Hussain, A. Iqbal, B. Yang, and A. Hussain, "Real time violence detection in surveillance videos using Convolutional Neural Networks," *Multimed. Tools Appl.*, vol. 81, no. 26, pp. 38151–38173, Nov. 2022, doi: 10.1007/s11042-022-13169-4.

[26] A.-M. R. Abdali and A.-T. Rana F., "Robust Real-Time Violence Detection in Video Using CNN And LSTM," *2019 2nd Sci. Conf. Comput. Sci.*, vol. 2, no. 1, pp. 104–108, 2019, doi: 10.1109/SCCS.2019.8852616.

[27] K. Gkountakos, K. Ioannidis, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Crowd Violence Detection from Video Footage," *Proc. - Int. Work. Content-Based Multimed. Index.*, vol. 2021-June, no. September, pp. 1-7, 2021, doi: 10.1109/CBMI50038.2021.9461921.