ISSN: 2093-5374 / E-ISSN: 2093-5382

Enhancing Real-Time Violence Detection in Video Surveillance Using Hybrid Deep Learning Model

Mohammed Inayathulla[1*], and K Rajasekhara Rao[2]

[1*]Research Scholar, Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh, India. 183030016.phd@gmail.com, https://orcid.org/0000-0001-9358-3687

[2]Professor, Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh, India. krr@kluniversity.ac.in, https://orcid.org/0000-0001-5904-7370

Received: November 05, 2024; Revised: December 17, 2024; Accepted: February 06, 2025; Published: March 31, 2025

Abstract

One of the crucial aspects of maintaining public safety and security in different environments is detecting violence in video surveillance (VS). Conventional systems are unable to accurately differentiate between violent and non-violent actions due to multi-factor nature and relative subtlety of violence, as well as environmental constraints. The use of advanced Deep Learning (DL) models, specifically Recurrent (NN) Neural Networks (RNN) and Convolutional NN (CNN), along with its types, such as ResNet, and bidirectional Long Short-Term Memory (Bi-LSTM) units, to address this problem. It serves as the focus of this research. To efficiently utilise both spatial (S) and temporal (T) data, the combination of ResNet50V2 architecture with bidirectional GRU and Bi-LSTM layers was employed by the suggested hybrid model. The model has a high success rate and much lower False Positives (FP) after being trained on a wide variety of real-world events. This model's computational efficiency and wide range of applications to various surveillance situations are also discussed, along with its potential for Real-Time (RT) operation. The DL architectures are an effective approach for creating VD systems that are reliable, adaptable, and scalable and it was demonstrated by the outcomes.

Keywords: Violence Detection (VD), Deep Learning (DL), Surveillance Systems, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs).

1 Introduction

The VD is a critical measure to enhance safety and security across different settings, from public places to personal facilities (Socha & Kogut, 2020). The ultimate goal of violence detection (VD) is the real-time identification of aggressive behavior or violence on various surveillance systems to facilitate prevention and timely elimination of further reoccurrences (Durães et al., 2021). This technological solution can help reduce risks, improve safety measures and protect human existence. Currently, growing adoption and complexity of security surveillance have contributed to the increased demand for automated VD systems (Alizadeh et al., 2020; Carniani et al., 2016). These systems should accurately

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 16, number: 1 (March), pp. 344-361. DOI: *10.58346/JOWUA.2025.I1.021*

*Corresponding author: Research Scholar, Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh, India.

identify violence from normal activities to minimize false positives and ensure efficient timely interventions (Mumtaz et al., 2023).

Automated VD poses numerous challenges despite its potential benefits. First, violence is a broad and multi-faceted phenomenon that does not always lend itself well to simple visual distinctions or mechanized classification. Different forms of violence present different visual signals at different intensities. In many cases, it would be mission impossible to differentiate all kinds of violent from non-violent behaviors based on visual information alone (Nikaeen et al., 2017). Moreover, variations in lighting conditions, camera angles, and visual obstructions can further complicate the task of accurately identifying violence (Ullah et al., 2023). In turn, the context of violence may vary considerably as well, wherein models should generalize well across distinct scenarios and not need to be retrained at each installation.

In addition, the complexity of human behavior also complicates violence detection (Wu et al., 2020; Priyadharshini & Balamurugan, 2023). Body language, movements and interaction can vary just enough to prevent successful identification of violent actions by traditional algorithms. A lot of current systems are based on hand-made feature extraction, that can have insufficient accuracy for the differences between violent and non-violent behavior (Sumithra & Sakshi, 2024). In addition, to achieve RT processing, a high level of computational efficiency must be ensured, which also requires additional restrictions on the effectiveness of such systems (Dang et al., 2020). Therefore, it is not an easy task to create robust and effective violence detection systems (Sharif et al., 2018).

Recent years have seen the rise of deep learning models as a potential avenue to overcome the aforementioned challenges (Burhan et al., 2023). Because they can be fed large amounts of data on which to base their decisions, these approaches have been proven to be more effective at detecting subtle differences between violent and non-violent activities (Sánchez et al., 2020). CNNs, RNNs, and its variants have demonstrated effectiveness in several kinds of computer vision (CV) tasks, such as VD, because of their performance in action relatedness and understanding. Using deep learning models allows the system to determine which elements of individual video frameworks are relevant and to employ them to diagnose violence.

Deep learning models have changed how violence detection works by allowing models to automatically learn hierarchical features from training data (Sahay et al., 2022). These models have become versatile and adaptable to diverse environments, making the best out of any novel context. Furthermore, hardware and software development have made it possible to run models in real-time applications, shortening response rates for security staff significantly. Therefore, deep learning model-based solution has been a critical development in the history of violence detection by providing a scalable and time-efficient solution to a critical security issue. The organization of the paper as follows: Section-II discuss the Literature survey and proposed model is explained in Section-III. The results are discussed in Section-IV (Chamyan & Farahani, 2016).

2 Literature

Tang et al., (2024) suggested a DL-based work to detect violence in video and images. The Faster R-CNN (FR-CNN) was adjusted to detect violent characteristics in cartoon and animation images, since cartoon and animation images are complicated and difficult to detect. The modified model should achieve the highest performance value. The backbone model of the suggested video and image is modified from an inner lateral connection to the modified ResNet model for Feature Extraction (FE) of the frames. The inner lateral connection was replaced with the modulated deformable convolutional

(MDC) layer for features such that Feature Map (FM) extraction and newly distributed attention modules are presented using DAM model to enhance the extraction performance. A multiscale Region of Interest (ROI) Align also adapted in the video to detect the level of violence with different video, and numerous ROIs are selected. The classification method was integrated with the detection model to detect the violence levels of an individual frame in a given frame. New approaches such as FR-CNN, Cascade R-CNN, yolov3-spp, SSD, FCOS, yolov5, and also the modified Faster R-CNN were used to detect violence in a few frame of the video. The comparison performed on each model performance and the improved FR-CNN is effective at detecting blood in the video and image real-time with high accuracy. This study can generate a new violent detection model in the virtual animation video which helps several platforms and government bodies to regulate the entertainment source.

Ehsan et al., (2024) introduced an unsupervised STAT in order to differentiate behaviors accurately and avoid the problem of insufficient violence data. The framework included a person detector, motion feature extractor, STAT network, and an output interpretation module. It worked well in a variety of environments by object detection (OD) in each frame and eliminating irrelevant background information. As the patterns of violent motion changed quickly with high velocity, T features were required to counteract such cognitive challenges, and such were employed as input to the STAT network. Hence, the STAT network, which had been trained on normal behavior data, converted the normal motion extremely inadequate into the spatial frame. The actions were categorised by comparing the actual and rebuilt frame and assessing the reconstruction error in the framework's output interpretation because the STAT network was unable to accurately rebuild violent frames due to the intricacy of violent behaviour. The suggested unsupervised approach achieved accuracy levels on par with prior works, thereby surpassing them in generality.

Mukto et al., (2024) developed an effective Crime Monitoring System (CMS) which can detect crime in real time using a camera surveillance system to alert the most responsible officer. This CMS was developed to counterbalance the weaknesses of human inattention, slow response, and slacking in detecting a crime, for example. The suggested CMS detects crime situations by fusing several DL methods and Image-Processing (IP) techniques with the features and mechanisms of closed-circuit television cameras. The three steps of the CMS dedicated weapon detection (WD), VD, and Face recognition (FR), operate in separate ways. To detect weapons, the transferred models are used, while the transferred models and face recognition algorithms are used to detect violence and recognize belongings. More specifically, the YOLOv5 model and MobileNetv2 were used to WD and VD, respectively; FR algorithms are used to recognize faces. Image and video datasets were used in the CMS, where the image dataset was used to test all the models and video datasets employed for training the VD model for MobileNet in these last models, which were tested using the frame-by-frame image dataset.

Park et al., (2024) presented a method to correctly model the spatiotemporal and causal characteristics of violent behaviors, using optical flow and RGB data. The method was based on a foundational network referred to as a Conv3D-based ResNet-3D model that processes High-Dimensional (HD) video data. Additionally, it included an attentional mechanism that activated the more critical frames in the RGB and optical-flow sequences more quickly than regular frames during violent episodes to enhance the efficiency and accuracy of the violence detection. The model was tested with UBI-Fight, Hockey, Crowd, and Movie fights datasets. Rendón-Segador et al., (2023) put forward a novel NN built on Vision Transformer and Neural Structured Learning that was trained with adversarial regularization. This network named, CrimeNet, showed that it substantially outperformed prior work while almost eliminating any cases of false positives. The results of the network’s examination using

the aforementioned four toughest violence-producing datasets: both binary and multi-class were presented.

Garcia-Cobo & SanMiguel, (2023) employed model took two other, more generic models – human pose extractors and change detectors – as inputs. The inputs were then combined using a novel technique based on summations rather than multiplications. Importantly, even when one of the two inputs brings a zero-value signal, the information can still be transferred. This feature made the new approach outperform the alternatives in the literature. Lastly, to incorporate both S and T information, a convolutional version of the standard LSTM, ConvLSTM, was used. The results of the conducted experiments on several benchmark datasets demonstrated that the suggested model was both accurate and efficient, achieving the (SOTA)state-of-the-art results while also requiring significantly fewer trainable parameters.

Singh et al.,)2020) designed the Automating Threat Recognition System using deep learning algorithms. It aimed to automatically detect signs of aggression and violence in RTby slicing out abnormalities from normalcy patterns. The deep learning models that would be employed in the study included CNN and RNN, which would recognize high movement levels in a frame and classify them according to the predispositions identified through multiple instances, meaning if an instance was identified in class x, it would be on the same level in the multiple instances loop. From the classifications rendered, a threat situation detection alert could be sounded. This means suspicious activities were taking place at that specific time.

Huszar et al., (2023) explored the potential of utilizing smart networks to generate the dynamic notions among actors and objects by using 3D convolutions (Conv) to carefully consider the spatial and segmental information of data. To improve the authenticity of VD in surveillance footage, the information learned from a pre-trained (AR) Action Recognition model was well exploited. The proposed methods were tested and analyzed with public datasets with substantial variations and complex testing content to evaluate its effectiveness.

3 Proposed Method

The proposed hybrid model contains ResNet50V2 as base model and Combined with Bidirectional GRU along with Bidirectional LSTM illustrated in Figure 1.

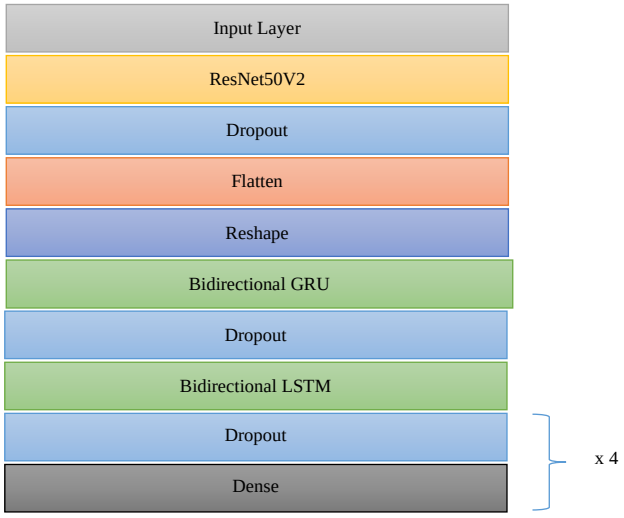

Figure 1: Proposed Method Architecture

- **Input Layer (IL)**

The "Input Layer" represents a point of entry for data into the neural network and thus is essential to determining the architectural basis for its functioning. For convolutional neural networks, such as ResNet50V2, this would be an initial step designed for the processing of raw image data. As such, it is defined by a variety of specifications closely aligned with features of input datasets at the time they reach the model. For example, when working with data on images containing red, green, and blue colors, one would expect to have an input layer consisting of three channels.

When the input data is received, the input layer serves as an interface, passing information through the subsequent layers of the network. The neurons within this layer correspond to a pixel or a feature of the input image, serving as the building blocks for more sophisticated representations in the layers that follow. The size of the IL plays a role in determining the size of the FM created by the Conv operations performed in the later layers. It means that the size and shape of this layer are adjusted to the network structure and data features.

Moreover, the input layer also enables the implementation of different pre-processing steps that might be used on the input data prior to the propagation of the information throughout the network. Specifically, this may include normalization, resizing, or augmentation in order to improve the model's ability to generalize and be robust. The use of such pre-processing within the input layer context allows the implementation of network-wide standards across various sources of input while simultaneously allowing for the adaptation to the input-specific needs.

- **ResNet50V2**

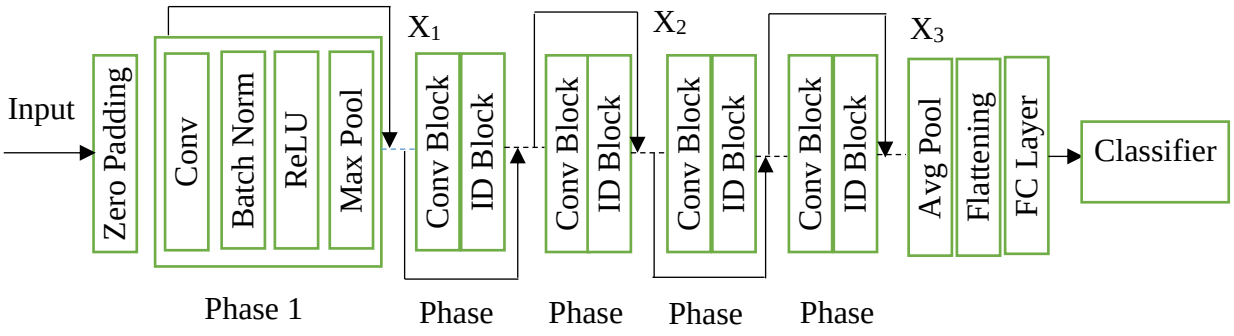

Figure 2: Resnet50V2 Architecture

- **ResNet50V2**: Is a convolutional neural network architecture with outstanding depth while performing quite efficiently when identifying images. Now, we can look into the architecture of ResNet50V2 and how its layers work and represented in Figure 2.

- **Input layer:** This is where the raw image data is fed into the ResNet50V2. This layer is usually set up to expects images of a specific size and a certain of channels. The size of the input images and the number of channels will determine the dimension of the layer.

- **Convolutional Layers (CL):** The main components of a ResNet50V2 are the CL, which are used at the beginning of the network to extract features from the input images. For the network to detect tiny patterns and structures in the image data, CLare necessary. To find characteristics like edges, textures, and patterns in the input image, Conv filters are applied. With every convolutional pass on the data, the

next layers in the channel learn to identify more complex and high-level (HL) representations of the input data.

A unique characteristic of ResNet50V2 is that the convolutional layers are organized into blocks. Each block consists of several convolutional layers, usually across different filter sizes and numbers. This hierarchy allows the network to learn features pertaining to different levels of abstractions. As a result, the higher the level of the layer, the more abstract the concept being learned, and a vice versa for the shallower layers which learn localized features. Multi-scale feature extraction is also critical in the network's spatial comprehension of the input images.

Furthermore, due to the inclusion of multiple convolutional layers within each block, more diverse features can be represented by the model. Rather than forcing a single layer to extract all important information within the input data, ResNet50V2 divides this responsibility among many layers within each block. As a result, the network not only has a higher ability to cover multiple features in many ways but also generates superior feature reuse and feature sharing throughout the network.

ResNet50V2: Residual blocks are a key novelty in ResNet50V2's architecture, marking the advancement of convolutional neural networks towards deeper and more efficient learning. The shortcuts, or skip connections as they are also known, are the most defining features of these blocks. They alter the blackbox approach of previous work by providing a shortcut for information flow. The optimization software may now need to push the weights of some layers to zero during training, in some instances, due to these layers' redundancy. These shortcuts allow networks to simply learn the residuals, hence the name "ResNet".

Residual blocks are essential to combat the problem of training very Deep NN (DNN). Specifically, as networks get deeper, they become hard to train due to problems associated with vanishing gradients (VG) and the degradation problem. This is where the accuracy of the network saturates, or sometimes, it even degenerates with increased depth. However, the residual blocks solve the vanishing gradient problem by providing paths through which identity of signals obtained before and after the convolutional layer is fed to the later layers to use.

Usually, each residual block includes a couple of convolutional layers, which act as the primary feature extracting section of the network. These layers also incorporate identity shortcut connections that just pass the input straight to the output without manipulation and form a shortcut. This structuring solution allows keeping the information from the prior layers intact for the next ones, making it easier for the network to notice and model subtle patterns and details in the data.

Batch Normalization (BN): A crucial method in many DNN, like ResNet50V2, is BN. It is employed for faster convergence and enhance training stability. Done after every convolutional layer, it ensures the normalization of the activations of the former layer in the mini-batch system loop while training is ongoing. The approach is important in reducing the internal covariate shift phenomena. In the case of the phenomenon, the distribution of the inputs in every layer shift because of training, thus making the network train at a slower rate and degrade.

Batch normalization makes activations centered around zero mean and standard deviation one to the layers that follow. This has the effect of stabilizing training because it ensures a more consistent and well-conditioned input to each layer, leading to smoother and faster convergence in the gradient descent (GD) optimization process. Additionally, batch normalization is a form of regularization, alleviating the need for dropout. During training, batch normalization introduces noise to the network, which has the benefit of reducing overfitting, environmental input patterns network by forcing the model to learn more generally useful patterns.

A key advantage of batch normalization is that it speeds up the training process. By effecting activation normalization in a mini-batch, the gradients passed down through the network have a greater tendency to vanish or explode much less, offering a more stable and faster training process. Networks with batch normalization technique may need fewer training epochs to get the same accuracy as networks without it.

- **Activation Functions (AF):** After every CL in ResNet50V2, AF, like the Rectified Linear Unit (ReLU), is applied. By leaving all positive values in the input unaltered and setting all negative values to zero, ReLU adds non-linearity to the network. This non-linear (NL)activation function is essential for CNNs to learn complex mappings between the input and output data. By introducing non-linearity, ReLU allows the network to model intricate patterns and relationships within the image data, enabling it to capture both simple and complex features effectively. Furthermore, by enabling gradients to flow more freely during backpropagation (BP), ReLU helps mitigate the VG problem (VGP), which can impede the training of DNN.

- **Pooling Layers:** In ResNet50V2, the task of downsampling the FM yielded by the CL is delegated to the pooling layers, which are most often implemented as max pooling: max pooling involves selecting the maximum value in a set of values at a predetermined spatial region. The pooling layers reduce the spatial dimensions of the feature maps and the network's computational complexity is also reduced. In addition, pooling layers reduce the risk of overfitting by decreasing the number of parameters, and they also make the network's learned representations more consistent to small changes and deformations of the input images. This latter property is especially important for object recognition, as it permits the network to learn to OD in images regardless of their precise, pixel-by-pixel location. By preserving only, the most informative parts of the input and discarding the rest, the pooling layers contribute to the network's robustness and generalization.

- **Fully connected (FC) layers:** Fully connected layers are a vital part of CNN's architecture such as ResNet50V2. Once the input image is processed by convolutional layers and feature maps are generated, the processed maps are reshaped back into a one-dimensional array called vectors. Flattening ensures that the spatial pattern and feature learnt are not disregarded in the following layers. This flattens the vector and provides input to the layers known as fully connected layers. They are also called dense layers achieving classification by domain knowledge in ResNet50V2 terminology. Dense layers contain neurons associated with each neuron of the previous layer. Fully connected layers can identify intricate relationships between learned characteristics and labels, making them useful for classification tasks. Fully connected layers in CNNs will find the best parameter set to derive data from the input images. In order to generate class probabilities, the final FC layers frequently employ a softmax AF.

In other words, the softmax normalizes the raw output scores of the previous layer to a probability distribution over different class. This means the output values are scaled from 0 to 1 and sum up to 1 and are therefore interpretable as probabilities. Each element of the output vector indicates the probability of the input belonging to that class. Hence, the softmax output is essential when the problem domain requires a measure of the model's confidence in assigning a class label to the input. Indeed, ResNet50V2 feeds the outcomes of the input image to the model, which uses softmax to produce the probability measure. The highest value indicates the categorization it perceives best fits the input image, thus demonstrating high performance.

ResNet50V2 is a Deep CNN (DCNN) architecture that excels at image recognition by combining residual blocks, BN, and skip connections (SC). Its hierarchical design enables it to extract progressively

intricate attributes from raw input data; it is popularly employed in tasks such as object detection, image classification (IC), and semantic segmentation.

- **Dropout**

An elementary regularization technique used in neural networks architectures, especially in deep learning, to avoid overfitting is dropout. Overfitting happens when a model learns the training data so well that it cannot generalize patterns, so it performs poorly on unseen data. Dropout tackles this barrier by randomly dropping or setting to zero a portion of the units/neurons in a layer during training. Therefore, it shuts down the units' connection by simply removing it, making the remaining units learn more mindful and autonomous features.

At each iteration during training, dropout randomly sets some pre-specified probability p of neurons' output to zero, in other words, it turns these neurons "off" during this particular training instance. This strategy prevents the neural network from becoming reliant on any one specific subset of features or neurons and rather forces it to obtain a more varied and diverse representation of the input. This is a type of regularizer or an ensemble method, since at test time, when applying the network to make predictions, dropout is turned off and the whole network with all neurons active is used. However, all learned internal weights are scaled by the dropout probability p such that the expected output remains constant. Therefore, in reality, the network is making predictions using multiple slightly different networks which share many common weights. Each network is obtained from the original by introducing a different dropout mask. Dropout thus regularizes the neural network by averaging over many models, and it also makes the model more robust by eliminating the danger of developing a complex, overfit model.

- **Flatten**

In a neural network, the Flatten layer (FL) is an essential part that can reshape the input tensor into a one-dimensional array which is also called a vector. In the framework of DL, data is used in the form of tensors which are multi-dimensional arrays. Tensors move through the various layers of a neural network and are changed and computed within those layers. However, before going through fully connected or dense layers and output layers, tensors often need to be flattened to work with these types of layers.

In summary, when a Flatten layer is added to a neural network architecture, it receives a multi-dimensional input tensor and flattens it along all dimensions except the batch dimension, transforming it into a one-dimensional array. It is crucial to note that while doing this, the total count of elements in the tensor remains the same, which implies that no information is lost, as it is only a matter of reshaping without changing the content. This way, the data becomes easily passable for the next layers, such as dense layers, that can only work with a uniform matrix or vector, making the processing step more efficient. Consequently, the FL acts as a connector for the CL and FC layers within the architecture by reshaping the input data and, in this manner, ensures the optimal information flowing, feature detection, and classification.

- **Reshape**

The reshape layer is another basic component that is required in deep learning architectures. As the name suggests, this layer enables one to reshape or change the grid dimensions of the input data while the total number of elements remains constant. It is largely used when the shape of the data should be changed to fit the demands of the following layers or operations in the network.

The reshape layer, on the other hand, simply takes an input tensor of any shape and reshapes it to a shape provided by the user. It means if you have an input tensor of shape, you can reshape it into any other shape such as or stretch it to the shape. This kind of freedom offered by a reshape layer allows neural network engineers to create more complicated models by converting the data into the desired format. By implementing the reshape layer in addition to other layers like fully connected and convolutional layers, the neural network engineer can build fascinating architectures that can handle different types of information and tasks.

- **Bidirectional GRU (Bi-GRU)**

A Bi- GRU is a RNN architecture that reads the input sequence in both directions: it processes information in forward and then backward order. By reading and processing the input in both directions, a Bidirectional GRU receives information from the past and future while aiming to enhance its understanding of patterns and relations existing in sequence sequences. Here, we will discuss more how a Bidirectional GRU processes information and its benefits.

To begin, let's break down the fundamental components of a GRU. An RNN is characterized by a special type of recurrent cell architecture that includes gating mechanisms designed to modulate the way data is transferred through the network. The most important gates in a GRU are the update gate and the reset gate. Gated mechanisms are responsible for deciding which information to retain and what to ignore at each time point, and as a result, the RNN can better capture long-term dependencies in sequences than traditional RNNs.

Introducing bidirectionality to a GRU, meanwhile, means that two separate GRU layers run at the same time in the opposite directions. The input sequence is read from the first to the last element in the forward pass.

In the backward pass, it is read from the last to the first element. Both of the passes have their hidden states and context is captured from both directions.

The forward pass occurs as follows: at every time step (TS), the forward GRU layer accepts input from the current TS and the hidden state (HS) from the previous TS. The layer computes the activations and updates its hidden state. The procedure is as follows during the backward pass: the backward GRU layer accepts input from the current TS and the HS from the next timestep. It computes representations and updates the hidden state Similarly.

In most cases, after both passes are done, the outputs of the forward and backward GRU layers are concatenated or combined in another way to form the final output sequence. Using both the past and future context to feed the model in turn would help it produce better quality predictions or classifications.

- **Bidirectional LSTM**

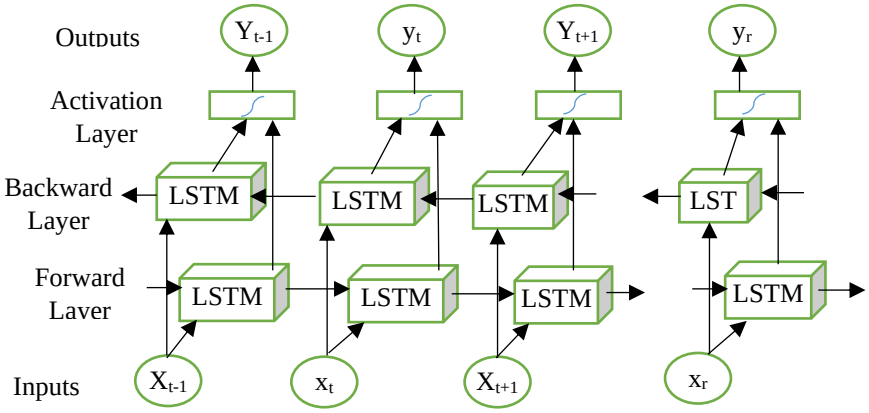

Figure 3: Architecture of Bidirectional LSTM

Figure 3 represents the Bidirectional LSTM architectures are a popular choice for sequence modeling systems, which are used in solving problems of natural language processing (NLP), correct recognition, and predicting time series. Standard LSTM sends the result in only one direction on each step of the recurrent connections of the network, often from the past and to the future. Although there are many areas in the real world where obtaining context not only from the past but also from the future can conveniently help in solving the problem.

Bi-LSTM overcomes this limitation by employing two different LSTM layers simultaneously, one to process the input sequence in the forward order and the second to process it in the reverse order. This feature helps the network understand the sequences in both the history and future contexts more clearly and model in the input sequences' dependencies.

Forward LSTM layer processes the input sequence one after the other from the beginning to the end at each time step. In contrast, the backward LSTM layer processes the sequence in reverse, starting from the last time step and heading towards the beginning. The hidden states that are output at each step of both LSTMs are then combined as the final status of that specific time step. When concatenated, the final status source information from both future and past contexts, hence the model becomes more unbiased than the usual uni-directional LSTMs.

One of the key advantages of bidirectional LSTMs is its capability to capture long-distance dependencies in sequence and context from both directions. This property of bidirectional LSTMs is crucial in sequence related tasks where one needs to consider the entire sequence, such as sentiment analysis, named entity recognition, and machine translation. Since bidirectional LSTMs can synthesize information from both past and future contexts, these bidirectional LSTMs can capture semantic relationships between tokens within the sequence, and hence they perform population in the various tasks involving sequence modeling.

- **Dense Layer**

A Dense Layer is a vital element used in Artificial NN (ANN), especially those models of deep learning. This component is otherwise called a fully connected layer – here, each neuron or node of a layer is completely connected to all the neurons of the previous or following layer. Consequently, the connections between input and output data become complex and closely related.

A Dense Layer basically does a linear operation on the input data then adds a NL AF. The linear operation is a dot product operation, where the input data is multiplied with a weight matrix and added with a bias vector. In this way, the network learns complex patterns in the dataset and their relationships. This is because the work of adjusting the weights is done during training through gradient descent method.

4 Experimental Results

This section discusses the detailed study of the results obtained from simulated work using our approach. The obtained dataset was collected from an open-source Kaggle. The approach was applied to this dataset. A total of 1000 videos on violence and 1000 of non-violence from YouTube appear in this dataset. The violence videos from the dataset are most of the real street fight videos from different places and situations. The non-violence videos are obtained from different human activities such as sports, eating, walking, among others.

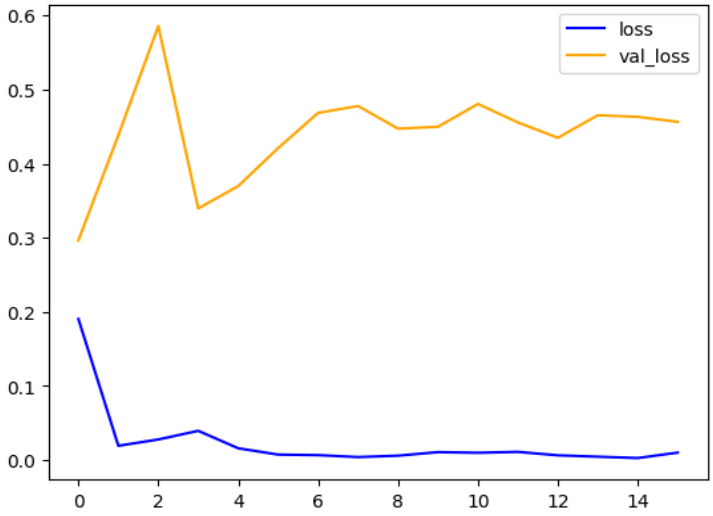

Figure 4: Training and Validation Loss

Figure 4 shows the Training Loss is the error measured on the training dataset when the model is being trained. Thus, training loss measures the performance of the model on the training data. As a measure of the discrepancy between the target and the model's predicted output, the loss function (LF) should be kept to a minimum. The model's ability to converge to the result that improves with decreasing training loss. Various kinds of LF, including hinge loss, cross-entropy (CE), and mean squared error (MSE), can be applied based on the kind of problem being addressed.

The term Validation Loss refers to the quantification of error computed on a distinct dataset known as the validation set. The validation set is separate from the training data and acts as an unbiased evaluation of the model's ability to perform on new, unknown data. Validation loss is a useful metric for measuring the performance of a model during training and acts as a guidance to avoid overfitting. Overfitting is the phenomenon when the model excessively focuses on memorizing the training data rather than developing a broad understanding, resulting in worse performance when applied to new, unknown data. Hence, closely monitoring and decreasing the validation loss is crucial to guarantee the model's efficacy in real-world situations.

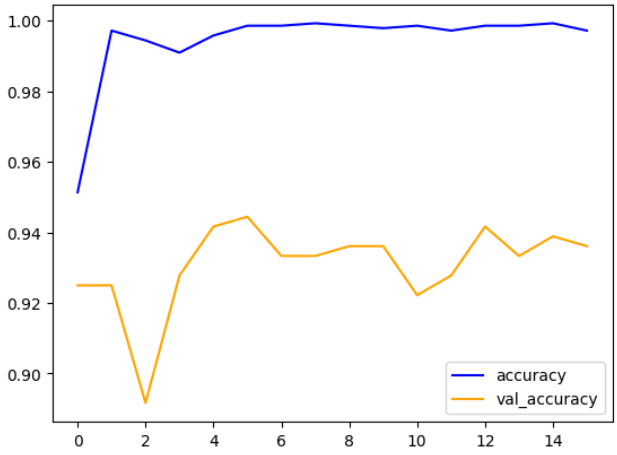

Figure 5: Training and Validation Accuracy

Figure 5 represents the Training Accuracy is a metric that evaluates the accuracy of classification of the training dataset as a proportion of correctly classified. It indicates the extent to which the model is learning from the training dataset. Nevertheless, high training accuracy is no assurance that the model will generalize well to new and unseen data, as the model may have memorized the training set with no real comprehension of the data-set patterns. In other words, while the training accuracy is critical to assess the model's learning, some validation performance is vital to address the new data.

Validation accuracy refers to the number of correctly labeled samples out of the entire validation dataset. It measures the model's capability in performing on data that it was not trained with, hence helps in generalizing its performance. Validation accuracy is an essential metric that has to be monitored throughout the training process. Practitioners need to evaluate the model by comparing training and validation accuracies and the losses on the validation data. The insights help them understand the model's performance during training and adjust the hyperparameters to avoid overfitting and enhance overall performance. The model has learnt the most pertinent patterns from the training data, and it can accurately classify new, unseen cases, according to high validation accuracy.

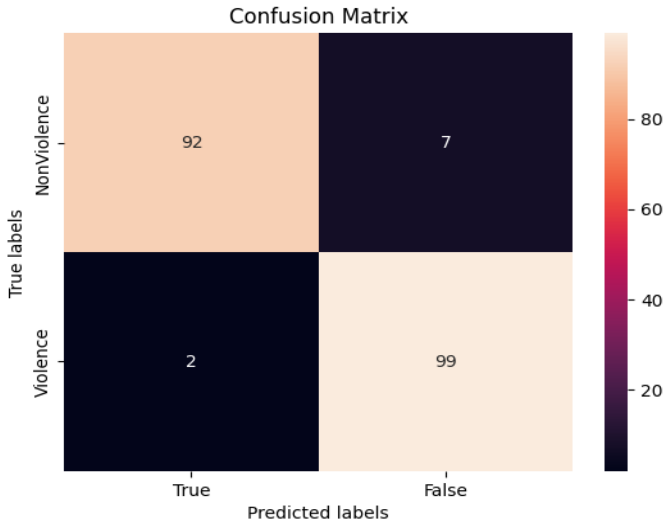

Figure 6: Confusion Matrix (CM)

Figure 6 shows CM. The given confusion matrix shows the results of the classification model where the task was to identify two classes: “Non-Violence” and “Violence.” In other words, each cell of a classification matrix illustrates the number of outcomes with the model making correct and incorrect predictions, considering the actual result. In the table, the rows display the true classes, and the columns show the predicted classes.

Having a glance at the upper-left cell, we are aware that 92 cases were properly classified as Non-Violence when in the upper-right cell, there were 7 “Non-Violence” misclassified as “Violence”. It can be observed that the false positive rate is 7 out of 99 “Non-Violence” cases. For the lower-left cell, we can identify that only 2 “Violence” cases were misclassified as “Non-Violence”. As a result, the false negative rate was low. Finally cell in lower-right cell shows 99 instances, which are rightfully classified as “Violence” by the model. These are instances when the model rightly predicted the label “Violence” and the actual label was also “Violence”. Altogether, the CM gives ideas of the model’s quality.

Table 1: Classification Report

	Precision (P)	**Recall (R)**	**F1-Score (F1-measure)**
Non-Violence	0.98	0.93	0.95
Violence	0.93	0.98	0.96
Accuracy	0.95		

Table 1shows Classification report. The performance metrics of the suggested model, which divides cases into two classes: "Non-Violence" and "Violence" are displayed in this table. It shows the total accuracy as well as the P, R, and F1-measure for each class. The percentage of real positive predictions among all positive predictions, is known as P.

While comparing the “Non-Violence” class, the precision is 0.98; as a result, approximately 98% of instances labeled as “Non-Violence” were indeed non-violent. Likewise, the precision of the “Violence” course is 0.93; hence, roughly 93% of instances predicted as “Violence” were true.

Recall, or sensitivity, captures the fraction of how many true positive instances the model has accurately predicted. In this case, “Non-Violence” was 0.93, indicating that the model has classified 93% of all non-violent instances. In contrast, the same ratio for “Violence” is 0.98, meaning that the model accounted for 98% of all violent cases. F1-score is the harmonic mean of P and R, as such, it gives equal weight to both measures while considering how their values are related to each other. It is especially useful when classes are imbalanced. In this light, F1-scores of Non-Violence and Violence are respectively 0.95 and 0.96 which means that the performance on entries is quite high for both classes.

Finally, the overall accuracy of the model is obtained as 0.95 and it means the proportion of the number of the correct-classified instances to the total instances is 0.95. An accuracy of 0.95 means that the model can differentiate effectively between the “Non-Violence” and “Violence” instances with high proportion of correctness.

4.1 Prediction for Frame by Frame

The Frames obtained from a video underwent the process by using proposed method and correctly predicted whether the violence exist in the frame or not that shows in Figure 7.

(a) Correctly predicted as Non Violence

(b) Correctly predicted as Non Violence

(c) Correctly predicted as Violence

(d) Correctly predicted as Violence

(e) Correctly predicted as Violence

(f) Correctly predicted as Non Violence

(g) Correctly predicted as Violence

(h) Correctly predicted as Violence

Figure 7: Prediction of Non Violence and Violence frame by frame

4.2 Prediction for Video

(a) Video 1 (correctly predicted as violence)

(b) Video 2 (correctly predicted as Non-violence)

Figure 8: Prediction of Non-Violence and Violence for Video

Figure 8 shows that the two videos above, the proposed method correctly predicted violence and Non- violence video. The video 1 correctly predicted as violence and confidence obtained is 0.99. Video 2 correctly predicted as Non-violence and get confidence value is 0.99 in table 2.

Table 2: Comparative Analysis

Method	Value
CNN	0.85
LSTM	0.89
ResNet50V2	0.91
Proposed hybrid ResNet50V2	0.95

5 Conclusion

The results of the study validated the proposed ResNet50V2 integrated with bidirectional GRU/LSTM to be an efficient hybrid model in the domain of video-based violence detection. This was proven by the model's high performance in accurately distinguishing violence from non-violence in different experimental setups, which outperformed the previous models. In addition, the study underscored the

limitations of real-time testing as well as model portability across different environments. However, more work should be directed towards improving model applicability by making it more generalizable and efficient to perform real-time processing with minimal computational requirements to be applied in various platforms.

References

[1] Socha, R., & Kogut, B. (2020). Urban video surveillance as a tool to improve security in public spaces. *Sustainability*, *12*(15), 6210. https://doi.org/10.3390/su12156210

[2] Durães, D., Marcondes, F. S., Gonçalves, F., Fonseca, J., Machado, J., & Novais, P. (2021). Detection violent behaviors: a survey. In *Ambient Intelligence–Software and Applications: 11th International Symposium on Ambient Intelligence* (pp. 106-116). Springer International Publishing.

[3] Burhan, I. M., Ali, Q. A., Hussein, I. S., & Jaleel, R. A. (2023). Mobile-computer Vision Model with Deep Learning for Testing Classification and Status of Flowers Images by using IoTs Devices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *14*(1), 82-94. https://doi.org/10.58346/JOWUA. 2023.I1.007

[4] Ullah, F. U. M., Obaidat, M. S., Ullah, A., Muhammad, K., Hijji, M., & Baik, S. W. (2023). A comprehensive review on vision-based violence detection in surveillance videos. *ACM Computing Surveys*, *55*(10), 1-44. https://doi.org/10.1145/3561971

[5] Wu, P., Liu, J., Shi, Y., Sun, Y., Shao, F., Wu, Z., & Yang, Z. (2020). Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16* (pp. 322-339). Springer International Publishing.

[6] Chamyan, R., & Farahani, A. V. (2016). Surveying multi cellular convertors controllers in power systems. *International Academic Journal of Science and Engineering*, 3(2), 33–42.

[7] Sánchez, F. L., Hupont, I., Tabik, S., & Herrera, F. (2020). Revisiting crowd behaviour analysis through deep learning: Taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects. *Information Fusion*, *64*, 318-335.

[8] Alizadeh, M., Andersson, K., & Schelen, O. (2020). A survey of secure internet of things in relation to blockchain. *Journal of Internet Services and Information Security (JISIS)*, *10*(3), 47-75. https://doi.org/10.22667/JISIS.2020.08.31.047

[9] Tang, Y., Chen, Y., Sharifuzzaman, S. A., & Li, T. (2024). An automatic fine-grained violence detection system for animation based on modified faster R-CNN. *Expert systems with applications*, *237*, 121691. https://doi.org/10.1016/j.eswa.2023.121691

[10] Sharif, A., Sher, A., Shouping, L., Kiran, A., & Sidra, F. (2018). Level of Education versus Level of Domestic Violence in Islamabad. *International Academic Journal of Innovative Research*, *5*(1), 91–108. https://doi.org/10.9756/IAJIR/V5I1/1810009

[11] Mukto, M. M., Hasan, M., Al Mahmud, M. M., Haque, I., Ahmed, M. A., Jabid, T., ... & Islam, M. (2024). Design of a real-time crime monitoring system using deep learning techniques. *Intelligent Systems with Applications*, *21*, 200311. https://doi.org/10.1016/j.iswa.2023.200311

[12] Priyadharshini, S., & Balamurugan, P. (2023, April). Empirical Analysis of Packet-loss and Content Modification based detection to secure Flying Ad-hoc Networks (FANETs). In *2023 International Conference on Networking and Communications (ICNWC)* (pp. 1-8). IEEE.

[13] Rendón-Segador, F. J., Álvarez-García, J. A., Salazar-González, J. L., & Tommasi, T. (2023). Crimenet: Neural structured learning using vision transformer for violence detection. *Neural networks*, *161*, 318-329.

[14] Carniani, E., Costantino, G., Marino, F., Martinelli, F., & Mori, P. (2016). Enhancing Video Surveillance with Usage Control and Privacy-Preserving Solutions. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *7*(4), 20-40.
[15] Singh, V., Singh, S., & Gupta, P. (2020). Real-time anomaly recognition through CCTV using neural networks. *Procedia Computer Science*, *173*, 254-263.
[16] Sumithra, S., & Sakshi, S. (2024). Exploring the Factors Influencing Usage Behavior of the Digital Library Remote Access (DLRA) Facility in a Private Higher Education Institution in India. *Indian Journal of Information Sources and Services*, *14*(1), 78–84. https://doi.org/10.51983/ijiss-2024.14.1.4033
[17] Mumtaz, N., Ejaz, N., Habib, S., Mohsin, S. M., Tiwari, P., Band, S. S., & Kumar, N. (2023). An overview of violence detection techniques: current challenges and future directions. *Artificial intelligence review*, *56*(5), 4641-4666.
[18] Dang, L. M., Min, K., Wang, H., Piran, M. J., Lee, C. H., & Moon, H. (2020). Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, *108*, 107561. https://doi.org/10.1016/j.patcog.2020.107561
[19] Sahay, K. B., Balachander, B., Jagadeesh, B., Kumar, G. A., Kumar, R., & Parvathy, L. R. (2022). A real time crime scene intelligent video surveillance system in violence detection framework using deep learning techniques. *Computers and Electrical Engineering*, *103*, 108319. https://doi.org/10.1016/j.compeleceng.2022.108319
[20] Ehsan, T. Z., Nahvi, M., & Mohtavipour, S. M. (2024). An accurate violence detection framework using unsupervised spatial–temporal action translation network. *The Visual Computer*, *40*(3), 1515-1535.
[21] Park, J. H., Mahmoud, M., & Kang, H. S. (2024). Conv3D-based video violence detection network using optical flow and RGB data. *Sensors*, *24*(2), 317. https://doi.org/10.3390/s24020317
[22] Garcia-Cobo, G., & SanMiguel, J. C. (2023). Human skeletons and change detection for efficient violence detection in surveillance videos. *Computer Vision and Image Understanding*, *233*, 103739. https://doi.org/10.1016/j.cviu.2023.103739
[23] Huszar, V. D., Adhikarla, V. K., Négyesi, I., & Krasznay, C. (2023). Toward fast and accurate violence detection for automated video surveillance applications. *IEEE Access*, *11*, 18772-18793. https://doi.org/10.1109/ACCESS.2023.3245521
[24] Nikaeen, Z., Dadaneh, S. Z., Navkhasi, J., & Nematzadeh, M. (2017). The relationship between the perceived value and satisfaction with the behavioral attitudes of the sport club customers. *International Academic Journal of Business Management*, 4(1), 236–244.

Authors Biography

Dr. Kurra Rajasekhara Rao, Pro-Vice Chancellor is a professor of Computer Science and Engineering (C.S.E.) having more than 35 years of teaching and research as well as administrative experience at KL University. His current research interests include topics related to Embedded Systems, Software Engineering, Software Testing, Data Sciences, Image Processing and Knowledge Management. He has authored a book and has more than 240 research publications in various International/National Journals and Conferences. Dr. KRR is a recognized as 'Research Guide' in many reputed universities and 32 doctorates were awarded under his guidance till now. Prior to this, he discharged duties in various organizations and as a Director at Usha Rama College of Engineering & Technology (Autonomous), Telaprolu, A.P. He contributed as a Member in Board of Studies for CSE & IT, at various prestigious institutions. Dr. KRR has been the Editor-in-Chief of an International Journal of Systems & Technologies (IJST), a renowned International Journal for 4 years. He is a Fellow of IETE, APAS. Life Member of IE, ISTE, ISCA and CSI.

Mohammed Inayathulla is a research scholar at KL University Vijayawada. He is having more than 10 years of teaching experience in Computer Science and Engineering in India. He has been awarded with Master's Degree in Computer Science and Engineering at Jawaharlal Nehru Technological University (JNTU) Anantapur. He has published 12 papers in various International / National Journals, Conferences. His area of interest includes Machine Learning, Deep Learning, Computer Vision. He is having professional membership in CSI and ISTE associations.