WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Automated Video Events Detection and Classification using CNN-GRU Model

Sajjad H. Hendi[1,*], Hazeem B. Taher[2] and Karim Q Hussein[3]

[1]Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Iraq
[2]The University of Thi-Qar – College of Education for Pure Sciences, Thi-Qar, Iraq
[3]Mustansiryha University-Faculty of Science, Computer Science Dept, Baghdad, Iraq

*Corresponding Author: Sajjad H. Hendi

DOI: https://doi.org/10.52866/ijcsm.0000.00.00.000
Received: September 2023; Accepted: November 2023; Available online: December 2023

ABSTRACT: In the era of vast and continuous video content creation, manually identifying crucial events becomes a tedious and inefficient task. To address this challenge, we propose a CNN-GRU model that automatically detects and classifies significant events in videos. This model employs ResNet50 Convolutional Neural Networks (CNNs) to extract visual features from video frames, followed by Gated Recurrent Units (GRUs) for temporal modelling and event recognition. By leveraging the sequential data handling capabilities of GRUs, our model captures temporal patterns across frames. We evaluate the model's performance using accuracy and F1-score metrics on the VIRAT dataset, containing 1,555 events across 12 event classes. Our approach achieves promising results, with an event classification accuracy of 75.22%.

Keywords: Deep learning, CNNs, GRUs, Video events, Keyframe

1. INTRODUCTION

In recent decades, there has been a notable surge in the worldwide prevalence of surveillance cameras, commonly referred to as Closed-circuit television (CCTV) systems. Given the extensive proliferation of cameras, the task of monitoring the copious amount of data generated by these devices poses a formidable challenge for human operators (Farrington et al., 2007).

Due to the rise of outdoor events, automatic event recognition technologies are needed. This capacity is important for event management, security, and tourism. The field has advanced with machine learning and deep learning. Computer vision and acoustic analysis have improved outdoor event recognition (Xu, 2021).

An outdoor human events recognition system necessitates the fulfilment of multiple requirements. Initially, the system must possess the capability to effectively manage substantial quantities of video data obtained from numerous cameras. Furthermore, the system must possess the capability to accurately identify and classify a diverse array of human activities within outdoor settings. Additionally, the system must possess the capability to function in real-time, thereby enabling the prompt delivery of alerts when deemed necessary (Sreenu & Durai, 2019).

The researchers used different and varied methods to Recognition outdoor events, starting from the traditional methods and then machine learning, up to deep learning [(Kamel et al., 2018), (Avola et al., 2021), (Pawar & Attar, 2019)]. Traditional video action recognition methods face significant challenges. They struggle to capture crucial long-range temporal patterns, hindering their ability to comprehend video dynamics. These approaches often fail to effectively incorporate temporal data, especially in untrimmed videos where actions occur briefly. Moreover, they are tailored for trimmed videos where actions occupy substantial time, making them less adept at detecting actions in realistic scenarios

*Corresponding author: phd202030562@iips.icci.edu.iq

where actions are fleeting [(Wang et al., 2016) ,8].

Handcrafted features are another limitation of traditional methods. While deep convolutional networks can autonomously learn meaningful visual representations, traditional techniques rely on manually designed features that may not encapsulate the intricate diversity of video actions [1].

Furthermore, these traditional approaches are computationally intensive, storage-demanding, and lack end-to-end trainability. They rely on a two-stage process involving pre-computed motion information, unlike deep convolutional networks that can learn motion representations directly from video frames [2].

Deep convolutional networks, along with newer strategies, offer promising solutions to address these limitations and attain state-of-the-art action recognition performance. To counter the challenge of capturing long-term temporal dynamics, 3D spatio-temporal convolutions have emerged as a natural extension of 2D CNNs for videos. However, most current 3D CNN methods focus solely on RGB input, ignoring valuable optical flow and depth data, limiting their ability to harness multimodal information [3, 4].

The need for substantial training datasets is another constraint of CNN algorithms [5, 6]. Although CNNs excel in still-image recognition with large datasets, video-based action detection suffers due to the scarcity of extensive video datasets, potentially leading to overfitting and reduced model generalization [7].

CNN approaches for video action detection were initially challenged in capturing long-term temporal connections, fully leveraging multimodal information, and effectively utilizing extensive training datasets. However, the introduction of Recurrent Neural Networks (RNNs) has effectively addressed these limitations, facilitating the advancement of video-based action recognition.

RNNs and their various iterations, notably Long Short-Term Memory (LSTM), have made noteworthy advancements in the field of temporal modelling for the purpose of human activity identification [8]. Recurrent Neural Networks (RNNs) possess the inherent capability to effectively retain and utilize preceding information within a sequence [9, 10]. However, Long Short-Term Memory (LSTM) networks specifically tackle the issue of vanishing gradients, hence enabling improved representation of extended temporal relationships [9, 11]. Nevertheless, the computational complexity associated with Long Short-Term Memory (LSTM) poses a significant obstacle, particularly when dealing with lengthy sequences and visual input of high dimensionality [9, 12]. In response to this matter, an alternative form of Recurrent Neural Networks (RNNs) known as Gated Recurrent Unit (GRU) has been suggested [13]. The GRU presents a more streamlined option that mitigates the computational load while effectively capturing extended interdependence [9]

Based on the preceding explanation, we present a novel model that addresses the aforementioned challenges. Our approach involves extracting features from the frames using a Convolutional Neural Network (CNN) with a customized layers arrangement. Subsequently, we employ a Gated Recurrent Unit (GRU) to capture the long-term dependencies in the data while minimizing computational complexity.

2. RELATED WORKS

Sultani at. el. [14] proposes an approach to learning anomalies in surveillance videos using weakly labelled training videos and deep multiple instances ranking framework. During the training process, the proposed method incorporates sparsity and temporal smoothness constraints into the ranking loss function in order to enhance the localization of anomalies. the proposed method used C3D for Feature extraction, while C3D features offer significant potential for video analysis but come with certain drawbacks. Their computation demands are high, leading to prolonged processing times, and they require substantial memory due to their 3D convolutional nature.

Jaouedi, at. el. [15] Introduced two methodologies for human action recognition. The initial motion detection and tracking method uses Gaussian Mixture Model (GMM) and Kalman filter. The suggested method identifies human motion in every video frame and tracks it using the Kalman filter. Deep-learning recurrent neural networks (RNN) are used in the second way to retain a conceptual state and understand movement. Gated Recurrent Units (GRUs) train the recurrent models to extract dataset features. The human action recognition methodology is then integrated to improve its performance on a larger dataset. Motion detection and tracking using GMM and Kalman filters work poorly in videos with cluttered backgrounds because background objects and movements can interfere with human action detection and tracking.

Amin Ullah at. el. [23] presented a theoretical structure for the identification and classification of activities inside surveillance footage obtained from industrial environments. The surveillance video stream is initially segmented into significant shots, with shot selection being performed through the utilization of a convolutional neural network (CNN) that incorporates human saliency attributes. Subsequently, the convolutional layers of a FlowNet2 CNN model are employed to extract temporal aspects of an activity within the sequence of frames. In this study, a multilayer LSTM model is proposed as a means of effectively capturing and learning long-term sequences within temporal optical flow data, with the ultimate goal of enhancing activity recognition. This approach encompasses a substantial level of computational intricacy and

necessitates substantial resources for its execution.

J.-O. Jeong [24] proposed a hybrid SlowFast network-YOLO model technique is used to recognize human activity in surveillance videos. the SlowFast network on annotated actions was trained and used YOLO's object detection to identify and locate activities in surveillance videos. The intricacy of video surveillance with varied camera perspectives makes accuracy challenging due to class disparities and the tiny scale of human beings. This work uses both models to improve video activity recognition by tackling dataset variation and precise localization. Training with pre-trained weights improves convergence speed. However, the mean average precision (mAP) achieved on the validation set is relatively low (around 0.1). Class imbalances and small human subjects in videos could contribute to this.

Hayat Ullah at. el. [25] In this study, a spatial-temporal cascaded framework is presented for human activity recognition. The framework is designed to be computationally efficient and versatile, utilizing deep discriminative spatial and temporal data. The proposed CNN architecture employs a dual attention mechanism that combines channel and spatial attention to capture significant human-centric features from video frames, thereby representing human behaviours. The convolutional and dual channel-spatial attention layers are designed to prioritize spatial receptive fields that contain objects within feature maps. Stacked bi-directional gated recurrent units (Bi-GRUs) employ a combination of forward and backward pass gradient learning in order to imitate long-term temporal patterns and human action recognition by leveraging discriminative salient information.

3. VIRAT VIDEO DATASET

A sizable video dataset gathered for research on object detection, object tracking, and event recognition is called the VIRAT Video Dataset Release 2.0_VIRAT Ground [26]. The VIRAT Ground Dataset is a subset of the VIRAT Video Dataset, which comprises a collection of ground-truth annotated video sequences for a variety of applications, including event identification, object tracking, and object detection.

This dataset consists of videos taken using stationary, high-definition cameras in eleven different situations. (1080p or 720p). The scenes are made up of multiple video clips, each of which could have one or more instances of events from eleven different categories. There are eleven videos in this collection, broken up into 329 individual video snippets. The categories of events and their corresponding shortcodes are displayed in Table 1.

Table 1. The types of events and the shortcode

N	Activity	Shortcode
1	A person loading an object onto a vehicle	LAV
2	A person unloading an object from a vehicle	UAV
3	The person opening a vehicle trunk	OAT
4	A person closing a vehicle trunk	CAT
5	The person getting into a vehicle	GIT
6	The person getting out of a vehicle	GOT
7	Person gesturing	GST
8	Not existing in Release 2.0	-
9	The person carrying an object	CAO
10	Person running	RUN
11	A person entering a facility	EAF
12	A person exiting a facility	XAF

The 1555 events annotated video clips from a variety of sensors and cameras, encompassing a wide range of outside settings, are included in the VIRAT Ground Dataset. Annotations for events are included in the dataset, which is helpful for creating and assessing computer vision systems. Samples from the VIRAT Video Dataset Release 2.0_VIRAT Ground Dataset are shown in Figure 1 [27].

4. PROPOSED METHODOLOGY

This section provides a detailed analysis of our proposed event recognition technique and its component parts. In order to aid understanding, the suggested approach is divided into two independent parts, each of which is covered separately. We proposed that features are extracted from input videos using 2D CNN architecture. Our second primary component is the RNN-GRU network. The benefit of using GRU in the process of recognizing events from videos lies in its ability to extract sequential data associated with events. Figure 2 illustrates the conceptual workflow of our suggested approach.

FIGURE 1. **Two samples from the VIRAT Video Dataset Release 2.0_VIRAT Ground Dataset.**

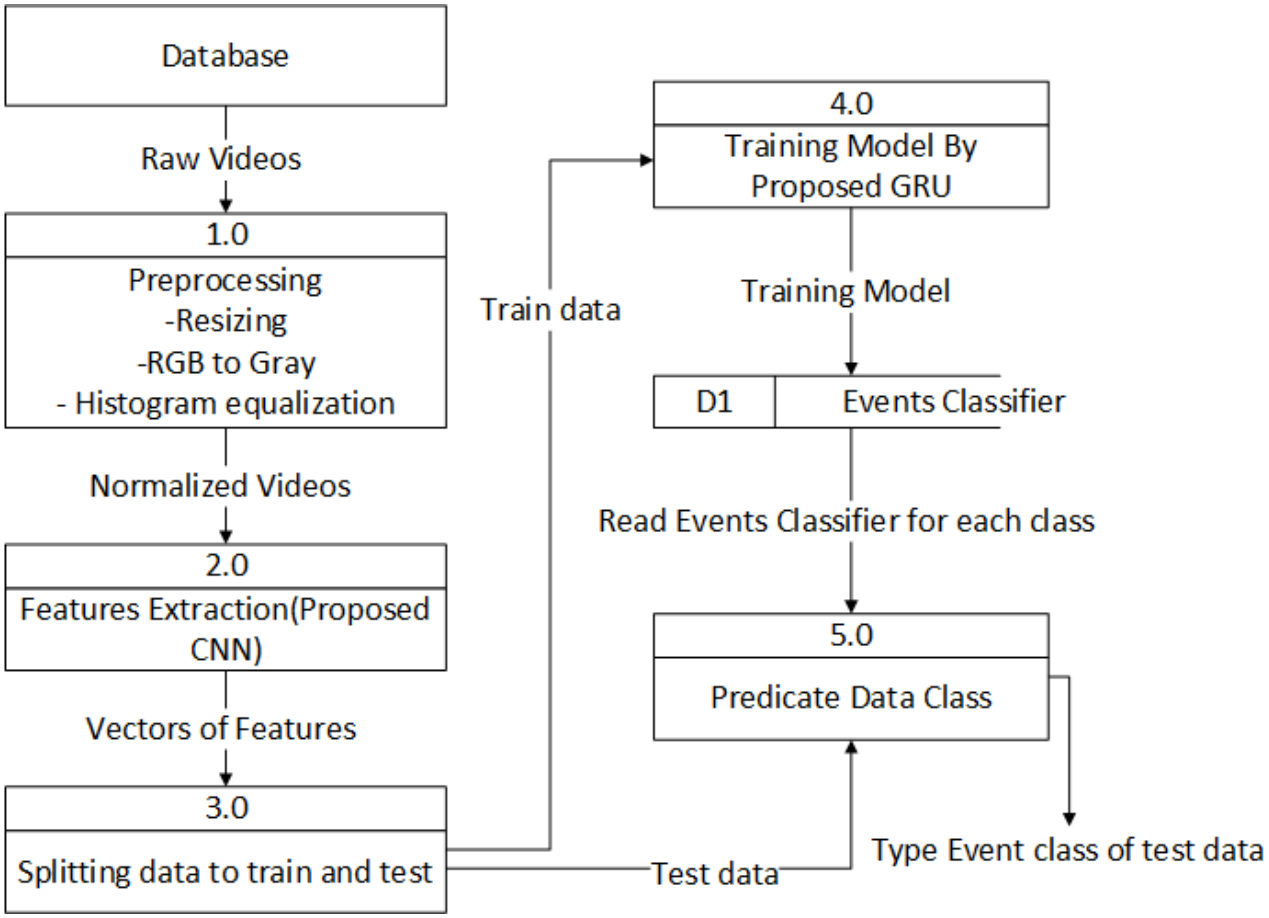

FIGURE 2. **Overview of the Proposed Model**

4.1 PREPROCESSING

The VIRAT Video Dataset Release 2.0_VIRAT Ground Dataset necessitates obtaining the dataset files from a credible source, ascertaining the format of the dataset, and loading the dataset by means of the applicable technique.

Several data analysis or machine learning libraries have been used to handle and analyse the data once the dataset has been loaded into memory. Python tools like NumPy, Pandas, and Scikit-learn may be utilised for preparing and analysing data. Pre-processing procedures are covered in the sections that follow.

4.1.1. Resizing

to guarantee that every frame in the movie has the same size and to lower the histogram equalization algorithm's computing demands. The frames for each video are then resized to a standard size of 224x224 [24].

4.1.2. Convert to Gray

Converting an RGB (Red, Green, Blue) image to a grayscale image is a common image processing task that involves representing the image using only shades of grey rather than full color. There are various methods for converting RGB to grayscale, but one of the most common methods is the luminosity method.

By using the luminosity approach, every RGB pixel in an image is transformed into a grayscale value according to how much of a contribution it makes to the overall perceived brightness of the image. The formula for converting RGB to grayscale using the luminosity method is:

$$Gray = 0.21R + 0.72G + 0.07B \tag{1}$$

This formula incorporates the consideration that the human visual system exhibits greater sensitivity to green light in comparison to red or blue light. The coefficients used in the formula are approximate and can be adjusted based on the specific requirements of the application. RGB image have convert to grayscale using the luminosity method in a video, we would need to apply the above formula to each frame of the video.

FIGURE 3. Converting a colorful frame into grey a) Original colored frame b) Gary's frame

4.1.3. Histogram equalization

To achieve a more equitable dispersion of pixel intensities across the dynamic range that is at the disposal the histogram equalization has been applied to the grayscale frame [7]. The first step is to calculate the histogram of the grayscale frame. The histogram represents the frequency of occurrence of each gray level in the image, as explained in Eq. (2).

$$PMF_n = \frac{I_n}{N}, n \in (0, 1, .., L-1) \tag{2}$$

Where PMF_n is the normalized histogram of gray level frame f with bins for each intensity n. I_n number of pixels with intensity n, N is the total number of pixels, L is the number of gray levels and 256 for 8-bit frames.

Then, calculate the cumulative distribution function (CDF):

$$CDF_j = \sum_{n=0}^{j} PMF_n \tag{3}$$

Normalize CDF values to cover all gray levels (0–L-1) by multiplying the CDF value by (L -1):

$$NCDF_j = (L-1) * CDF_j \tag{4}$$

Calculate the gray level mapping function. Map function:

$$G_j = round(NCDF_j) \tag{5}$$

The variable G_j denotes the gray level that has been equalized for the initial gray level j. The round function is utilized for the purpose of rounding numerical values to their nearest integer values. After applying this transformation to every pixel in the image, the intensity histogram of the image will be more uniformly distributed. This can result in an image with better contrast and improved visual appearance. Figure 4 shows the one sample frame before applying Histogram equalization and after applying Histogram equalization.

FIGURE 4. The Gray frame and its histogram a) Original frame b) Equalized frame

4.2 FEATURES EXTRACTION USING 2DCNNS

Convolutional Neural Networks (CNNs) are among the most commonly used methods for image recognition and feature extraction. The input frame is taken and subsequently processed in order to extract the relevant features from each frame. The Convolutional Neural Network (CNN) model was implemented utilizing Keras, a free open-source deep learning library written in Python. For each pre-processed frame, a sequence of convolutional layers, employing filters (Kernels), Pooling, and fully connected (FC) configuration layers, will be utilized to successfully extract features. In ResNet50 CNN model, all parameters required for initialization are outlined in Table 2, with random values assigned. We have worked to eliminate the model's last layer of classification. This indicates that rather than producing class probabilities, the model will output the features that were retrieved from the last convolutional layer.

In computer vision and deep learning, the use of 2D-CNNs for feature extraction from movies is a commonly used method. CNNs have shown to be successful at processing pictures, and with little adjustments, they may also be used to handle movies. I'll go over the main procedures for utilizing CNNs to extract features from videos in this context. Pre-processing the video frames by reducing them to a set size and normalizing the pixel values is the first stage in feature extraction using 2D CNNs. This stage guarantees consistency in the input data and lowers the CNN model's processing cost.

Table 2. Initialization Parameters in Convolutional Neural Networks (CNNs)

Parameters	Values
Activation function	ReLU, Linear
Filter kernel sizes	3
Gradient Descent Optimizer	Adam
Stride	1
Padding	Same
Learning rate	0.0001

Next, a 2D CNN model is trained on the pre-processed video frames to extract relevant features. CNN layers include convolutional, pooling, and Convolutional layers that filter input frames to extract spatial features while pooling layers downsample feature maps to minimize their dimensionality. The extracted features were used as input to the gated recurrent units (GRUs).

In a 2D CNN, the feature extraction layers are the convolutional layers and the pooling layers. These layers work together to learn hierarchical spatial features from the input data (video frames, in our case). Convolutional layers apply filters (also called kernels) to the input, capturing local patterns within the data. As they go deeper into the network, the convolutional layers learn increasingly complex and high-level features. Pooling layers, such as max-pooling or average-pooling, help reduce the spatial dimensions of the feature maps and control the number of parameters in the network, making the model more efficient and robust to overfitting.

4.3 CLASSIFICATION USING GRUS

The sequence model using GRUs for video classification was implemented, which is a task of predicting the class label of a video based on its content. The model takes as input a sequence of features extracted from video frames using 2D CNNs and uses GRUs to model the temporal dynamics of the video by processing the sequence of features sequentially.

Table 3. The proposed GRU layers

Layer	Description
Input Layers	The input layers of the model are two: frame_features_input(a sequence of features extracted from video frames) mask_input(a binary mask that indicates which elements of the input sequence are valid and which are padded).
GRU Layers	There are two Gated Recurrent Unit (GRU) layers in the model. First layer: 16-units, responsibility of handling the input sequences while taking the binary mask into account. Second layer: 8-units, processes the outputs that the preceding layer produced.
Dropout Layer	After the second GRU layer, a dropout layer is added to reduce the possibility of overfitting. dropout rate: 0.4
Fully Connected Layer	A fully connected layer with 8 units and a rectified linear unit (ReLU) activation function is added after the dropout layer.
Output Layer	The output layer (dense layer) that creates a probability distribution across the different class labels using a softmax activation function. The number of units in this layer is equal to the total number of unique class labels in the dataset.
Compilation	The loss function, optimizer, and evaluation measure for training and evaluation are configured using the compile () method of the Keras model. When there are several classes and the names are integers, the sparse_categorical_crossentropy loss function is employed, which is advantageous. The Adam optimizer is a popular and efficient optimizer that is used for gradient-based optimization. The accuracy value serves as a gauge for the model's performance.

The model architecture is composed of two GRU layers with dropout and fully linked layers, followed by a SoftMax output layer. The model generates a probability distribution across the class labels after receiving a collection of features that were taken from video frames together with a binary mask that indicates whether items in the sequence are correct. The model is compiled using a suitable loss function, optimizer, and evaluation metric for the given classification task.

5. EXPERIMENTS

In this section, we offer a thorough experimental assessment of our proposed model designed for event recognition in surveillance videos recorded outside buildings. We evaluate the effectiveness of our suggested model by analyzing its performance. First, we detail the implementation aspects and performance evaluation metrics used in this model. Next, we briefly touch upon the datasets employed for benchmarking experiments. Following that, we compare our proposed model with the state-of-the-art in event recognition for outdoor surveillance videos, considering each dataset involved in the experiments. Finally, we showcase visualizations of event recognition in surveillance footage taken outside buildings and perform a runtime analysis of our proposed method for real-time event recognition.

5.1 IMPLEMENTATION OVERVIEW

The following points describe the most important characteristics of the applied environment. Implementation: Windows 10, Python 3.10, TensorFlow 2.0, and on a computing system with an Intel core-i7 7th gen processor.

- Main memory:32GB.
- GPU: NVIDIA GeForce GTX 1080 with 8 GB graphics random-access memory.
- Data Split: 70% training and 30% validation.
- Training: 100 epochs, batch size of 64, Adam optimizer, learning rate of 0.0001.
- Loss Function: Categorical cross-entropy loss for weight adjustment.
- Sequence Learning: 20 frames sequence length without overlapping in GRU layers.
- Performance Evaluation: Accuracy and runtime metrics, comparison with contemporary methods.

5.2 RESULTS AND COMPARISONS

Table 5 shows model performance metrics on the test dataset. The following is a detailed analysis of each metric:

Table 4. The performance metrics of the proposed model

Metric	Score
Test Accuracy	75.22%
Test Precision	74.58%
Test Recall	75.22%
Test F1 Score	73.73%

The test accuracy metric denotes the proportion of accurately classified instances within the test dataset. The model attained a precision rate of 74.58%, indicating its ability to accurately forecast the class labels for 75.22% of the test samples. Figure 5 shows the accuracy and loss of the model.

In the context of testing, precision refers to the capacity of a model to accurately recognize positive instances among all instances that are predicted as positive. The calculation involves determining the proportion of true positives in relation to the combined total of true positives and false positives. The precision of the model was determined to be 74.58%, signifying the accuracy of positive sample predictions made by the model to be 74.58%.

The evaluation metric of Test Recall, which is also referred to as sensitivity or true positive rate, assesses the model's capacity to accurately recognize positive instances among all the existing positive instances. The calculation involves determining the ratio between the number of true positives and the sum of true positives and false negatives. With a recall rate of 75.22%, the model demonstrated its ability to correctly identify 75.22% of the actual positive cases.

Performance of the classification model is measured by the F1 score. The model's predictions are assessed using the harmonic mean of accuracy and recall. The evaluation metric provides a comprehensive assessment of the model's performance by accounting for both accuracy and recall. To get a given value, multiply the precision and recall product by two and divide the result by the total of the precision and recall. In this case, the model achieved an F1 score of 73.73%.

In general, the model performed satisfactorily on the test dataset. It is clear from the accuracy rate of 75.22% that the model is capable of accurately classifying a sizable percentage of the samples. The accuracy and recall scores, both above 70%, demonstrate the model's good performance in identifying positive cases and lowering false positives. The F1 score of 73.73% indicates that there may be room to improve the model's effectiveness by indicating a possible trade-off between recall and precision.

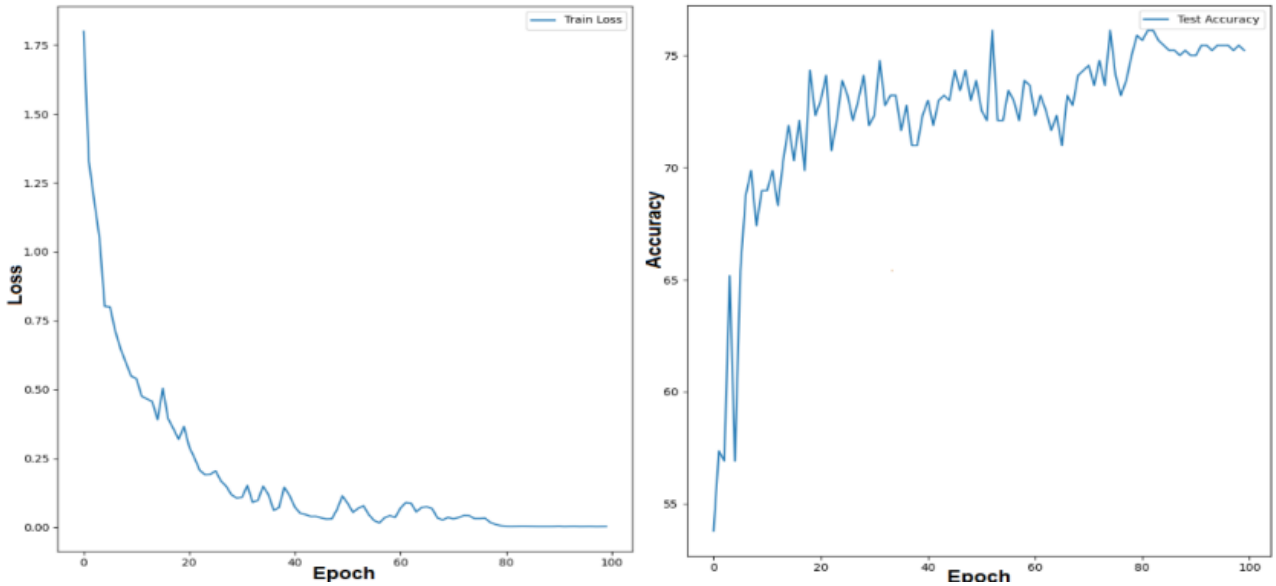

FIGURE 5. The accuracy and loss of the proposed model

It is essential to place the previously described results in the context of the specific problem and dataset. To determine potential pathways for improvement, more analysis and empirical study might be conducted. Some ideas include optimizing hyperparameters, exploring different architectures, or expanding the size of the training dataset. Ultimately, after assessing the different approaches used to pinpoint specific events in the VIRAT Video Dataset Release 2.0_VIRAT Ground Dataset, it was found that, in comparison to the previously used approaches, the suggested model produced results that were more accurate, as Table 6 illustrates. The model under consideration was subjected to comparative analysis with four established algorithms, as shown in Table 6. CAM [28], DHCM [29], ResNet50, and InceptionV3 have been trained on a large-scale dataset imagenet to learn a rich set of visual features.

Table 5. Comparisons of the proposed model to other models

Method	Accuracy
InceptionV3 + GRU	53.19%
DHCM [29]	66.8%
CAM [28]	62.9%
ResNet50 + GRU	71.94%
Proposed Model	75.22%

6. CONCLUSION AND FUTURE WORKS

This study presents a novel hybrid methodology for the identification and classification of noteworthy events in videos. The proposed approach leverages Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to achieve this objective. The model being proposed utilizes a combination of convolutional neural networks (CNNs) for visual feature extraction and gated recurrent units (GRUs) for temporal modelling and event recognition. The technique was evaluated using the VIRAT dataset, which consists of a wide range of events distributed across different categories. The empirical results show that the level of performance was good, with an accuracy of 75% in event categorization. This observation demonstrates how well the suggested methodology works to find noteworthy events in outside environments. In general, although the model that has been suggested displays promising, additional investigation and enhancement are required to progress event recognition systems with regard to dataset expansion, transfer learning, multimodal fusion, real-time detection, robustness, and human-centric recognition.

FUNDING

None

ACKNOWLEDGEMENT

I am deeply indebted to Informatics Institute for Graduate Studies and all the individuals who have played a role in my education and research. Their contributions and support have been invaluable, and I am honored to have been a part of the Informatics Institute for Graduate Studies academic community.

CONFLICTS OF INTEREST

The authors assert that they do not possess any identifiable conflicting financial interests or personal affiliations that may have potentially influenced the findings presented in this research article.

REFERENCES

[1] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
[2] Y. Du and L. Wang *Hierarchical recurrent neural network for skeleton based action recognition*, 2015.
[3] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1691–1703, 2012.
[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2015.
[5] P. Wang, P. Wang, S. Wang, Y. Hou, and W. Li *Skeleton-based action recognition using lstm and cnn*, 2017.
[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio *Empirical evaluation of gated recurrent neural networks on sequence modeling*, 2014.
[7] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," *Proc. of the IEEE conf. on computer vision and pattern recognition*, pp. 6479–6488, 2018.
[8] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "Deep learning approach for human action recognition using gated recurrent unit neural networks and motion analysis," *Journal of Computer Science*, vol. 15, no. 7, pp. 1040–1049, 2019.
[9] A. Ullah, K. Muhammad, J. Ser, S. W. Baik, and V. H. C. D. Albuquerque, "Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 9692–9702, 2019.
[10] J. O. Jeong *Human Activity Recognition with Computer Vision*.
[11] H. Ullah and A. Munir, "Human Activity Recognition Using Cascaded Dual Attention CNN and Bi-Directional GRU Framework," *Journal of Imaging*, vol. 9, no. 7, pp. 2023–2023.
[12] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. C. Chen, J. T. Lee, and . . & M Desai, "A large-scale benchmark dataset for event recognition in surveillance video," *CVPR*, pp. 3153–3160, 2011.
[13] P. Nair, U. Kumar, and S. Nandan, "COVID-19 Social Distance Surveillance Using Deep Learning," in *Computer Vision and Image Processing: 6th International Conf. Rupnagar, India, Revised Selected Papers, Part II*, pp. 288–298, Springer International Publishing, 2021.
[14] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video"," *Proc. of the IEEE conf. on computer vision and pattern recognition*, pp. 2491–2498, 2013.
[15] X. Wang and Q. Ji, "A hierarchical context model for event recognition in surveillance video," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2561–2568, 2014.
[16] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2740–2755, 2019.
[17] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *Computer Vision - ACCV*, vol. 2018, pp. 363–378, 2019.
[18] Meng, H.-Y. Xu-Hong, W.-H. Shi, and Shang, "Analysis of basketball technical movements based on human-computer interaction with deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–7, 2022.
[19] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2018.
[20] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2017.
[21] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
[22] B. Chen, H. Tang, Z. Zhang, G. Tong, and B. Li, "Video-based action recognition using spurious-3d residual attention networks," *IET Image Processing*, vol. 16, pp. 3097–3111, 2022.