frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 16 October 2023
DOI 10.3389/fcomp.2023.1274928

OPEN ACCESS

EDITED BY
Changxin Gao,
Huazhong University of Science and Technology, China

REVIEWED BY
Xiang Wang,
Huazhong University of Science and Technology, China
Leyuan Liu,
Central China Normal University, China

*CORRESPONDENCE
Pham The Bao
ptbao@sgu.edu.vn

RECEIVED 09 August 2023
ACCEPTED 25 September 2023
PUBLISHED 16 October 2023

CITATION
Ha ND, Tran NY, Thuy LNL, Shimizu I and Bao PT (2023) Violence region localization in video and the school violent actions classification. *Front. Comput. Sci.* 5:1274928. doi: 10.3389/fcomp.2023.1274928

Violence region localization in video and the school violent actions classification

Ngo Duong Ha[1,2,3], Nhu Y. Tran[2], Le Nhi Lam Thuy[4], Ikuko Shimizu[5] and Pham The Bao[4]*

[1]Faculty of Mathematics and Computer Science, University of Science, Ho Chi Minh City, Vietnam, [2]Information Technology Faculty, Ho Chi Minh City University of Industry and Trade, Ho Chi Minh City, Vietnam, [3]Vietnam National University-Ho Chi Minh City, Ho Chi Minh City, Vietnam, [4]Information Science Faculty, Saigon University, Ho Chi Minh City, Vietnam, [5]Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, Tokyo, Japan

Classification of school violence has been proven to be an effective solution for preventing violence within educational institutions. As a result, technical proposals aimed at enhancing the efficacy of violence classification are of considerable interest to researchers. This study explores the utilization of the SORT tracking method for localizing and tracking objects in videos related to school violence, coupled with the application of LSTM and GRU methods to enhance the accuracy of the violence classification model. Furthermore, we introduce the concept of a padding box to localize, identify actions, and recover tracked objects lost during video playback. The integration of these techniques offers a robust and efficient system for analyzing and preventing violence in educational environments. The results demonstrate that object localization and recovery algorithms yield improved violent classification outcomes compared to both the SORT tracking and violence classification algorithms alone, achieving an impressive accuracy rate of 72.13%. These experimental findings hold promise, especially in educational settings, where the assumption of camera stability is justifiable. This distinction is crucial due to the unique characteristics of violence in educational environments, setting it apart from other forms of violence.

KEYWORDS

violence detection, GRU, LSTM, SORT, VGG

1. Introduction

The timely prevention of human actions and behaviors relies heavily on video surveillance, which plays a crucial role in ensuring public safety, especially in sports, schools, and other public places. To improve this safety, there is a significant focus on developing violence detection systems. These systems include identifying the position joints of objects to assess violent behavior (Pang et al., 2014; Lee et al., 2016) by detecting frames, joints' velocity and position of violent acts. Another application is the surveillance video-based violence detection (Souza et al., 2010; Bilinski and Bremond, 2016), as well as violence detection in football fields (Wen and Liu, 2008; Dinesh et al., 2019). Being able to observe and predict violent acts is crucial to ensuring safety, and the violence classification system can help rate the degree of violent acts, thereby contributing to problem-solving in various aspects of life.

Kinect sensor-based approach (Han et al., 2013; Pang et al., 2014; Lee et al., 2016) uses Cartesian coordinates to calculate the position of joints. From that, they can analyze joints to find out the violent acts. For example, if a person moves his hand quickly and connects to another's body, it will be interpreted as punching.

In optical flow approach, each pixel corresponds to the optical flow for motion vector and it can estimate the motion state of objects in the image sequence. Hence, the corresponding motion information is extracted from the optical flow field to detect abnormal behavior

(Huang and Chen, 2014; Zhang et al., 2015; Gao et al., 2016; Naik and Gopalakrishna, 2016; Wang Q. et al., 2016; Mahmoodi and Salajeghe, 2019). In the article (Saif and Mahayuddin, 2020) violence acts is detected by using optical flow to calculate angles and linear distances in each frame. Violent decisions are determined by classifier, each feature vector is generated from a set of probabilities for each classifier.

The correlation filters approach is a powerful tool in digital signal processing. The Kalman filter method is a powerful filter processed in many different problems (Wen and Liu, 2008; Shehzed et al., 2019). Kalman filter implements the regression method with noisy input sequences, in order to optimize the estimate value of the system. While the Particle filter method is based on the Hidden Markov Model (Klein et al., 2010). It means that the Hidden Markov Model plays a key role. The more realistic the model is, the more accurately the Bayesian solution can estimate the state of the object. The group of authors led by Liu has demonstrated that employing sparse decomposition in dense scenes is a crucial procedure for enhancing the tracking performance of occluded targets (Liu et al., 2023).

CNNs have demonstrated potential in violence recognition, yet they exhibit several limitations attributed to their primary design for static images. These constraints encompass an incapability to handle temporal information, a limited grasp of contextual nuances, and difficulties processing lengthy video content. Additionally, they confront challenges in coping with the variability in violence appearances. Mitigating these shortcomings may entail exploring hybrid models, such as CNN-RNN combinations, alongside leveraging diverse datasets and meticulous architectural designs. Striking a balance between the advantages of CNNs' feature extraction and the complexity of temporal analysis is pivotal for achieving effective violence recognition in video content. Recent advancements have seen violence recognition systems evolve into end-to-end deep learning models, employing robust techniques that amalgamate CNN and LSTM. These novel approaches streamline the feature extraction process inherent in conventional recognition systems, thereby enabling the learning of feature vectors during training (Fang et al., 2016; Wang L. et al., 2016; Zhou et al., 2017; Ramzan et al., 2019; Biswas et al., 2020; Roy et al., 2020; Ullah et al., 2021; Ye et al., 2021). For instance, Sudhakaran and Lanz introduced an end-to-end trainable deep neural network model, which comprises a convolutional neural network for extracting frame-level features, followed by feature aggregation in the temporal domain via convolutional long short-term memory (Sudhakaran and Lanz, 2017). Min-seok Kang proposed the incorporation of spatiotemporal attention modules and a frame-grouping methodology to construct a practical violence detection system. Spatial attention utilizes MSM to identify salient regions derived from motion boundaries, while temporal attention employs the T-SE block to recalibrate temporal features using a minimal number of additional parameters. This innovative pipeline has yielded significant performance enhancements (Kang et al., 2021). In another notable development (Zhou, 2022), authors introduced an inventive end-to-end model that combines the Transformer for human pose estimation with a 3D Convolutional Neural Network designed to capture spatial-temporal motion.

The process of classifying violent acts is a challenging one due to multiple factors that need to be addressed. One significant challenge is determining the precise object area involved in the violent act, even when the object is localized. In cases where information is missing, the object description must be broadly defined. Additionally, variations in lighting, camera movement, and background images can impact the classification of violent acts. For instance, when there are multiple objects present, it becomes necessary to distinguish between those involved in violent acts and those that are not. The analysis of features is crucial for the accurate classification of violent acts. Despite these challenges, there is currently no algorithm available that can accurately identify and classify all cases of violent acts.

The article presents a novel approach to classifying school violence by combining the SORT method (Bewley et al., 2016) with RNN methods like LSTM and GRU (Hochreiter and Schmidhuber, 1997; Cho et al., 2014). Our proposed method utilizes a padding box to localize and track objects, allowing for the extraction of features that improve the accuracy of violent act predictions. Additionally, we address the issue of lost tracked objects in the video and propose a solution to ensure that no information is lost during the object tracking process. The proposed solution of using a padding box to localize objects and object restoration based on the lost track on the SORT+VGG16+RNN algorithm resulted in an ∼10% increase in accuracy.

In Part 2 of the article, we present our work related to the Kalman filter (Kalman, 1960) combined with the Hungarian technique (Kuhn, 1955) in the SORT method. We propose using a padding box to localize objects and acts, as well as recover lost objects, all in service of classifying violence by combining SORT and RNN methods. We present experimental results in Section 3 and offer our conclusions and suggestions for future work in Section 4.

2. Method

2.1. SORT method

The SORT (Bewley et al., 2016) method focuses on the connection issue between the Kalman filter (Kalman, 1960) and the Hungarian algorithm (Kuhn, 1955). First, detect objects from frames by using the FrRCNN (VGG16) technique (Ren et al., 2017). Next, find a way to associate the bounding boxes collected in each frame and assign an ID to each object. Specifically,

- Detect: locate the objects in frame.
- Predict: Predict new locations of objects based on previous frames.
- Associate: Associate detected locations with predictable locations to assign the corresponding ID.

Detailed interpretation of the SORT algorithm flowchart in Figure 1:

Prediction: Predict the next state of each tracked object using a motion model. This typically involves updating the object's position and velocity based on the previous state.

Detection: Obtain the detections from the current frame. These can be obtained using

a detection algorithm or a separate object detection model.

Data Association: Assign each detection to one of the existing tracked objects or mark it as a new object. To accomplish this, compute the similarity between each detection and each tracked object. This can be done using methods like the Intersection over Union (IoU) or the Hungarian algorithm.

Update: Update the state of each tracked object using the associated detection. This typically involves updating the object's position, velocity, and any other relevant attributes.

Create New Tracks: For any unmatched detection, create a new track and initialize it with the detection's information.

Track Management: Perform track management tasks such as track deletion, track merging, and track birth to maintain a manageable number of active tracks.

2.2. VGG network model

The VGG16 architecture, introduced at the time, stands out as one of the largest and most computationally expensive CNNs, boasting ~138 million parameters. Despite its resource-intensive nature, it has garnered exceptional results on benchmark datasets, notably the ImageNet Large Scale Visual Recognition Challenge, achieving a top-5 error rate of 7.5%. Overall, the VGG16 architecture serves as a formidable CNN model for image classification, characterized by its straightforwardness and hierarchical structure, facilitating comprehension and implementation, albeit demanding substantial computational resources for training and utilization.

VGG16—a convolutional neural network architecture consisting of 16 layers, 13 of which are convolutional layers with a

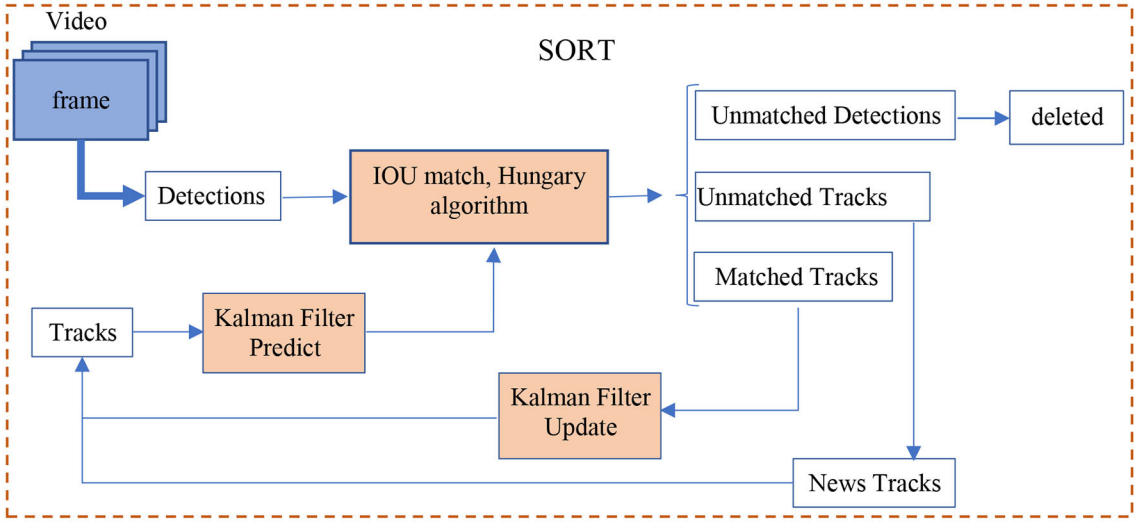

FIGURE 1
SORT algorithm flowchart (Bewley et al., 2016).

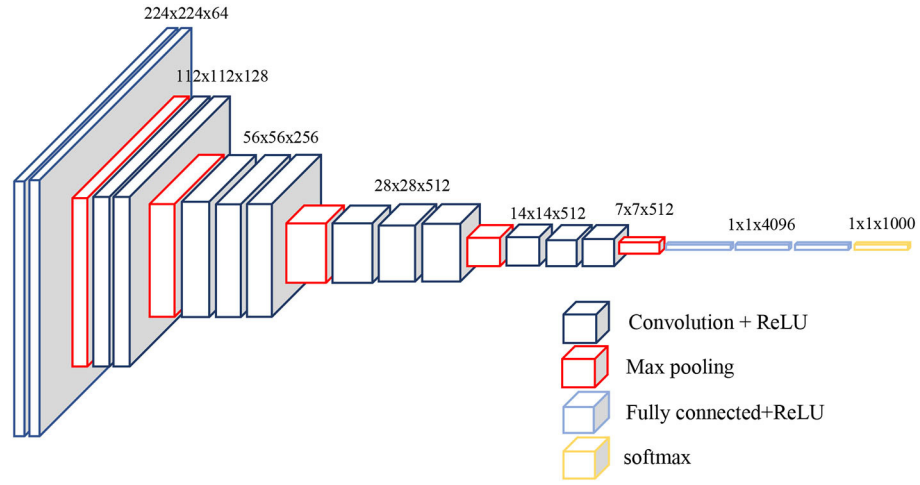

FIGURE 2
VGG16 network model (Simonyan and Zisserman, 2014).

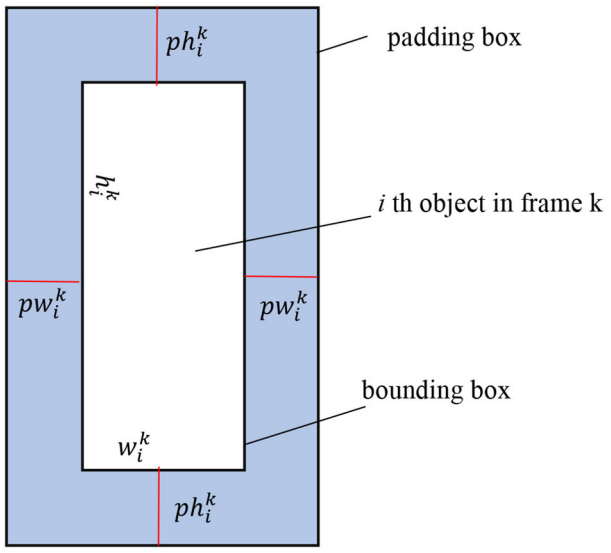

FIGURE 3
Padding box of *i* th object in frame k.

kernel size of 3 × 3 and the remaining three are fully connected—achieved an accuracy of 92.7% on the ImageNet dataset, which contains 14 million images from 1,000 different classes in Figure 2 (Simonyan and Zisserman, 2014).

2.3. LSTM network model

The Long Short-Term Memory (LSTM) network has been used in image processing through the incorporation of spatial attention mechanisms. Spatial attention allows the network to selectively focus on certain parts of an image, enabling it to effectively ignore irrelevant information and improve its performance on tasks such as object recognition, segmentation, video analysis, and captioning. LSTM network is composed of multiple interconnected LSTM cells. The LSTM block diagram consists of a cell c and three gates—the forget gate f, the input gate i, and the output gate o—which regulate the flow of information into and out of the cell. At each time step t, the gates receive input × and h hidden values from the previous time step t-1's output (Hochreiter and Schmidhuber, 1997).

LSTM's memory cells and gating mechanisms allow for the retention of important information over extended periods of time, enhancing the accuracy and flexibility of its sequential data modeling. It is also designed to overcome the vanishing gradient problem found in traditional RNNs, making it more efficient and effective during training. The training process of an LSTM network involves optimizing the gate and memory cell weights to improve network performance, often using backpropagation through time to propagate the error signal over multiple time steps. These networks have been widely applied in video analysis, speech recognition, and image captioning, leveraging their ability

Input: Frame $(k)^{th}$ (I_k) and the state array, its name is $Track_{k-1} = \{ id, x_{k-1}, y_{k-1}, \omega_{k-1}, h_{k-1}\}$ is information of object in the $(k\text{-}1)^{th}$ frame. ε is the threshold to choice or not choice an object to update.
Output: The state array at frame $(k)^{th}$, its name is $Track_k = \{ id, x_k, y_k, \omega_k, h_k\}$.

Step 1:
//Using Kalman Filter predict object state in frame $(k)^{th}$ based on $Track_{k-1}$
KF = KalmanFunction(I_k, $Track_{k-1}$);
//Using SORT method to extract the temp state array in frame $(k)^{th}$ with KF
$Temp\_Track_i$ = SORT(KF, I_k, $Track_{k-1}$);

Step 2:
// with trained FrRCNN-VGG16 model to detect objects in (I_k)
Objects = FrRCNN_VGG16(I_k);
// building the cost matrix by IOU based the difference values between the object states
// of the $Temp\_Track_i$ and the detected objects by FrRCNN-VGG16 model
Cost_matrix = buildMatrix(Objects, $Temp\_Track_i$);

Step 3:
// Using Hungarian method to solve cost matrix
Solution = HungarianFunction(Cost_matrix);

Step 4:
// Matching objects in detected objects (by FrRCNN-VGG 16 model) and objects in //$Track_{i-1}$
resultMatch = matchObjects(Solution, Objects, $Track_{i-1}$);

Step 5: Create an array of object states for the $Track_i$ in the frame k.

Step 5.1:
// update states of detected objects (by FrRCNN-VGG 16 model) by Kalman filter
updatedObjects = updateStates(Objects, $Track_{i-1}$, KF, resultMatch);

Step 5.2:
// If the object is unmatched in $Track_{i-1}$, then create a new Track and insert it into $Track_i$
for(k in Objects)
if(k $\notin$ $Track_{i-1}$) then $Track_i$ = $Track_i$ $\cup$ k;
endIf
endFor

Step 5.3:
//Check the object status in the track in the k−1 image with the object state in the detection. If the object state in the track in the k−1 image does not exist and $\beta < \varepsilon$, update the object state with track in the k image
for(k in $Track_{i-1}$)
if(k $\notin$ Objects) then
// computing β by formula (2).
$\beta = \sum (p_{kt} - p_{k-\ 1t})$
if($\beta < \varepsilon$) then $Track_i$ = $Track_i$ $\cup$ k;
endIf
endIf
endFor

Algorithm 1. The improved SORT algorithm.

to handle long-term dependencies and avoid the vanishing gradient problem to achieve remarkable results.

2.4. GRU network model

GRU is a type of recurrent neural network (RNN) that was introduced by Cho et al. GRU network consists of a set of recurrent layers, each of which has a set of internal gates that control the flow of information. These gates allow the GRU to selectively retain or discard information from previous time steps, which can help address the vanishing gradient problem that can occur with traditional RNNs. The GRU model can be trained using backpropagation through time, which involves computing the gradients of the loss with respect to the parameters at each time step and propagating them backwards through the network (Cho et al., 2014).

GRU networks are known for their simpler structure and fewer parameters, which make them more straightforward and quicker to train. The gating mechanisms in GRU networks are capable of selectively updating and forgetting information, allowing the network to handle long-term dependencies and generate accurate predictions. As a result, they have been successfully utilized in a wide range of applications such as natural language processing, speech recognition, and image captioning. Given these advantages, the GRU network is considered a highly promising architecture for modeling sequential data and continues to be a popular subject of research in the field of deep learning.

3. Proposed method

Problem: Locate the i^{th} object (with $i \in [1, n]$) at coordinates $(x_i^1, y_i^1)$and dimension (w_i^1, h_i^1) in the first image. Filter the state $(x_i^k, y_i^k, w_i^k, h_i^k)$ of the *i*th object in the subsequent frames. Based on the features of the object regions (excluding the background) in the k^{th} image and subsequent images from *(k+1)* to *(k+t)*, locate the violent classification in the sequence of object regions' features from the k^{th} to $(k+t)^{\text{th}}$ frames in the video.

When an object is involved in violent acts, the parts causing violence include hands, legs, and weapons. Therefore, in the next section, we propose using a padding box to locate the object and actions. Additionally, the object detection process in the time interval of camera frames may experience loss due to inaccuracies in the algorithm. Hence, we propose a method to recover the unknown object-tracking in the time interval.

3.1. Object localization

For the videos we separate from the data folder that the camera is relatively stable, when the object moves, the determination of the bounding box by the SORT (Bewley et al., 2016) algorithm does not contain the entire object area (including arms, legs, and weapons) engaging in violent acts. Therefore, we create a padding box containing the bounding box of the tracked object as shown in Figure 3. In addition, during the movement of the object near or far from the camera, we will adjust in the direction of increasing or decreasing the padding box, respectively.

First, we create a padding box based on the bounding box of the *i* object in the *k* frame and the weight pair (pw_i^k, ph_i^k) as shown in Figure 1. Next, in the process of object movement through each frame, the object near or far from the camera will, respectively, adjust in the direction of increasing or decreasing the size (pw_i^k, ph_i^k). Of the padding box according to the Formula (1).

$$pw_i^k = \frac{w_i^k}{w_i^{k-1}}^{*} pw_i^{k-1} \text{and } ph_i^k = \frac{h_i^k}{h_i^{k-1}}^{*} ph_i^{k-1} \tag{1}$$

In which, w_i^k and h_i^k are, respectively, the width and height dimensions of the bounding box containing the i-th object based on the track in frame *k*.

```
Input: Clips is array of videos; each item is
classified violence video with label.
Output: Model of classification of school
violence acts and results of an assessment of
school violence classification.
 Step 1:
  for(i in Clips)
   for(frame_k in Clips_i)
    // using Algorithm 1
    Track_ik =getTrack(frame_k, Track_i(k−1));
   endFor
  endFor
 Step 2:
  // Update the object state in the track in the
kth image by using the padding box with
   //formula (1) and based on the track in the
image k-1.
  for(i in Track)
   updateTrack(Track_i, formula(1));
  endFor
 Step 3:
  // Remove the background in the kth image
based on the array of object states in the
   //track in the k^th image.
  for(i in Clips)
   for(frame_k in Clips_i)
    // using Algorithm 1
    frame_k = removeBackground(frame_k, Track);
   endFor
  endFor
 Step 4:
  // extract features of Clips by VGG
  features = VGG_model(Clips);
 Step 5:
  RNN.initial(parameters, features);
  RNN.train();
  RNN.test();
```

Algorithm 2. Improved SORT_RNN.

3.2. Recovering object localization in time interval

Each object exists for a certain time t when it moves along a specific trajectory before leaving the camera monitoring. In time t, the object-tracking Kalman filter algorithm can lose track of the object and reappear the object signal after time t. Based on the object information before losing track, we proceed to recover the bounding box of that object in the next frames until the threshold is not satisfied according to Formula (2) (Anh et al., 2012).

$$\beta(X^{k-1}, X^k) = \sum_i \left(\left|pX_i^{k-1} - pX_i^k\right|\right) \quad (2)$$

In which, X^k is the bounding box of the X object in the kth frame, $pX_i^k = \frac{nX_i^k}{nX^k}$, with $i \in [0,\ 255]$ is gray level in histogram, nX_i^k is the number of pixels at the i gray level in X^k, nX^k is the number of pixels X^k.

Suppose that we lose track of the object being tracked in the kth image. In this case, we need to recover the object's information in that image. We can do this by calculating the similarity $\beta(X^{k-1}, X^k)$ between the region of the *k-1* image (X^{k-1}) and the region of the kth image (X^k), using Formula (2). If $\beta(X^{k-1}, X^k)$ is less than a chosen threshold ε, then the two image regions have high similarity (Anh et al., 2012), and we can recover the padding boxes of the object in the *k-1* image and vice versa.

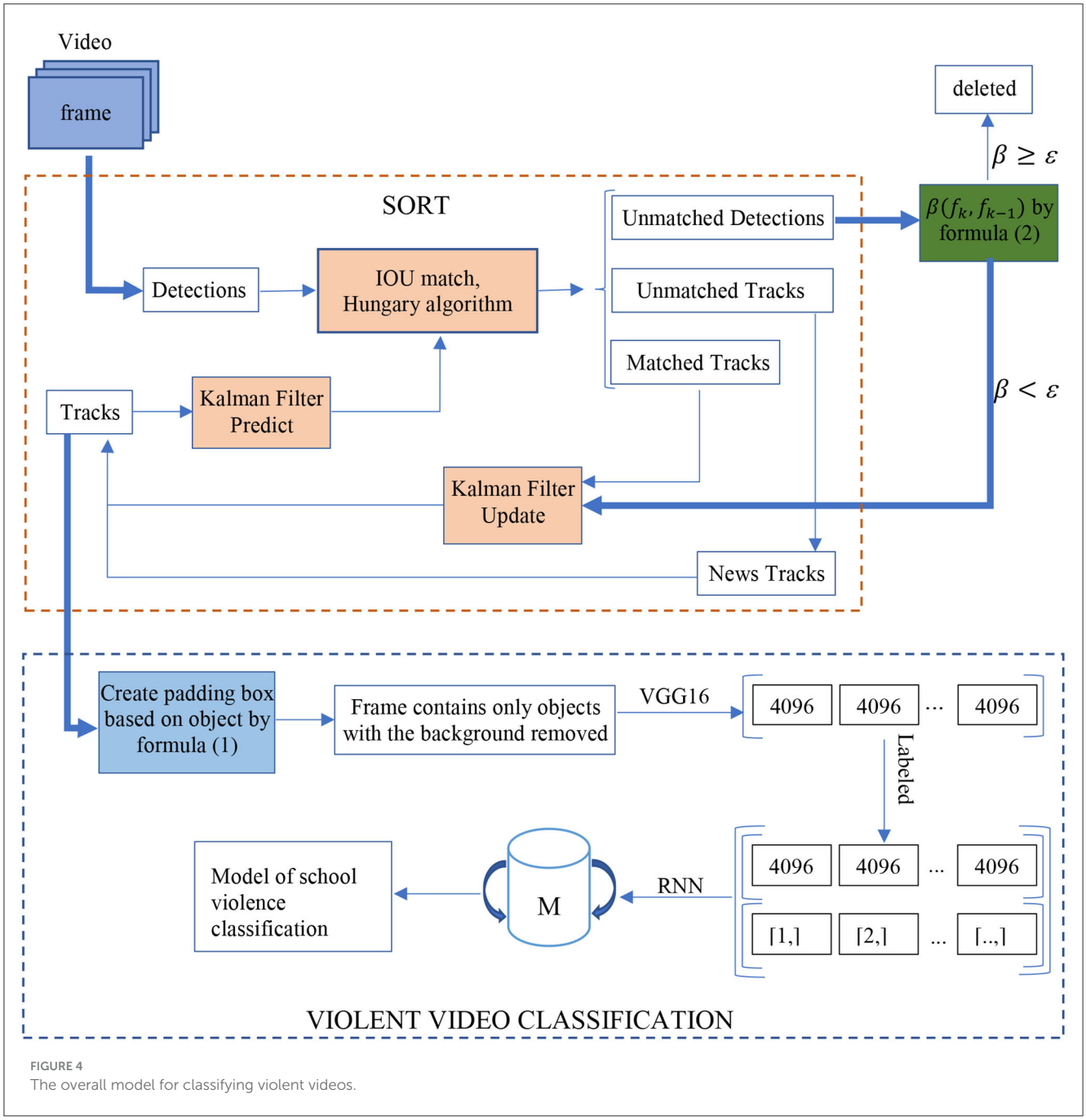

FIGURE 4
The overall model for classifying violent videos.

Our proposal is to enhance object detection in video using the SORT algorithm (Bewley et al., 2016). The algorithm's steps for processing an image are detailed in Algorithm 1, which is based on Figure 2.

3.3. Developing an algorithm to classify school violence

To detect and track objects within a video and classify instances of violence, we utilize the SORT method as described by Bewley et al. (2016). Specifically, this method combines the powerful VGG16 architecture with recurrent neural networks such as LSTM and GRU, which were first introduced by Hochreiter and Schmidhuber (1997) and later improved upon by Cho et al. (2014).

By implementing the SORT method, we can effectively identify and track objects within each frame of the video and use recurrent neural networks to analyze their movements and patterns. This allows us to accurately classify instances of violence within the video based on the movements and interactions of the objects in each frame. Additionally, the SORT method has been shown to have a high level of accuracy in object detection and tracking, making

FIGURE 5
Images of violent actions in schools.

it a reliable approach for analyzing videos with complex scenes and multiple objects. This is particularly important in the context of violence detection, where the movements and interactions of multiple individuals within a scene can be complex and difficult to analyze.

Moreover, the use of recurrent neural networks such as LSTM and GRU in the SORT method allows for a more comprehensive analysis of object movements over time. These networks can retain and utilize information from previous frames in their analysis, thereby enabling us to better understand the patterns and behaviors of objects within a video and make more accurate classifications of instances of violence. However, despite its advantages, this method has limitations in terms of efficiency.

To address this, we introduce the improved SORT_RNN algorithm in Algorithm 2, which is based on Figure 4. This algorithm proposes enhancements to the SORT+VGG16+LSTM method, which combines object detection and tracking with recurrent neural networks, with the aim of improving the efficiency of violence classification. The first step is Object Localization: Utilizing the SORT technique to locate and track objects in each frame of the video. During the SORT process, a padding box is generated, encapsulating the bounding box of the tracked object using Formula (1). Additionally, object localization is restored in the time interval using Formula (2). The second step is Feature Extraction: Applying the VGG16 model to extract features from the regions identified by SORT in each frame. The third step is LSTM Model Construction: Building an LSTM network to create a violence classification model. Finally, Violence Classification: Using the trained LSTM model to classify the frames in the video. In summary, these enhancements include incorporating object localization and recovery over time, which enables us to more accurately analyze the movements and interactions of objects within a video. By utilizing these enhancements, we can more efficiently and accurately classify instances of violence within a video.

4. Experimental result

4.1. Environment

4.1.1. Installation environment

We tested on a computer using 64-bit Windows 10 Home operating system, 16GB RAM, 6GB GPU, Intel Core (TM) chip i7-9750H CPU @ 2.6GHz. Python programming language version 3.6.2.

4.2. Data set

School violence can be classified into 17 different categories related to health consequences, as shown in Figure 5.

Level 1: Intimidation

- Pointing (L11): The act of aggressively pointing the finger at someone's face.

TABLE 1 Statistics by classification of school violence.

No	Code	Description	Number of videos	Average of frames/video
1	L11	Pointing	402	30.3
2	L12	Grab T-shirt collar	403	32.8
3	L13	Martial arts stance	401	30.1
4	L14	Pushing	404	29.0
5	L15	Pulling	400	29.7
6	L21	Neck clamping	402	30.3
7	L22	Strangling	402	30.6
8	L23	Taking hold of hair	404	30.1
9	L24	Hits	401	28.7
10	L25	Kicks	403	28.9
11	L26	Knee kicks	405	28.6
12	L27	Struggling	405	29.8
13	L28	Throwing objects	402	29.4
14	L29	Hit by material	402	28.2
15	L31	Undressing	401	31.7
16	L32	Victim lying on the floor	401	33.0
17	L33	Bleeding	401	30.0
18	L01	Non-violent	402	29.9
Average (μ)			402.3	30.6

- Grabbing a shirt collar (L12): The act of taking hold of someone's shirt collar.
- Martial arts stance (L13): The act of holding up both hands, bending forward, keeping a defensive posture against a person's attack.
- Pushing (L14): The act of using hand(s) to shove a person away.
- Pulling (L15): Two people stand on both sides of the victim, clamping the victim's hands.

Level 2: Non-life-threatening violence includes

- Neck clamping (L21): Wrapping an arm around victim's neck and pressing down.
- Strangling (L22): The act of using hands to choke another person's neck when standing or sitting in front of that person.
- Taking hold of hair (L23): The act of grabbing another person's hair aggressively.
- Hits (L24): The act of using a hand or arm to hit another person.
- Kicks (L25): The act of using a leg or foot to kick another person.
- Knee strike (L26): The act of raising the knee to kick another person, typically using a hand to grab another person.
- Struggling (L27): The act of holding a victim down with both hands.
- Throwing objects (L28): The act of throwing objects at a victim.
- Hit by material (L29): The act of hitting another person with an object, such as a stick or a chair.

Level 3: Life-threatening violence

- Undressing (L31): The act of taking off another person's clothing.
- Victim lying on the floor (L32): victim lying on the floor after being attacked.
- Bleeding (L33): The victim bleeds on the body after the attack.

Using a dataset of 7,240 clips (224 × 224 pixels) related to 18 kinds of school violence at VSiSGU. Statistics of clip results on each category through Table 1 and Figure 6.

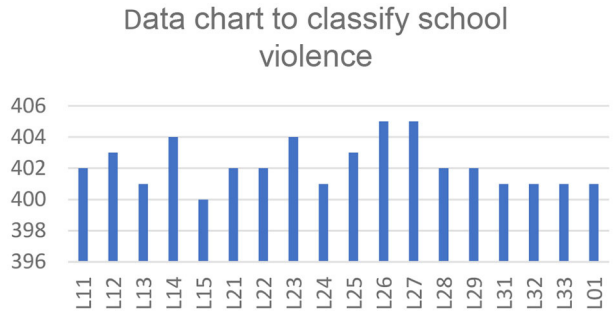

FIGURE 6
Clip data chart to classify school violence.

4.3. Result

Results of Figures 7–13 classifying violence based on VSiSGU dataset with our proposed improved SORT+LSTM, improved SORT+GRU algorithms and SORT+LSTM, SORT+GRU, VGG16+LSTM, VGG16+GRU, and VGG16+SimpleRNN algorithms. The results of the comparison between the algorithms are presented in Table 2.

Based on Figures 11–13, as well as Table 2, it can be observed that among the VGG16-based architectures, VGG16+LSTM achieves a higher accuracy of 71.11% with a loss of 0.0230. It outperforms VGG16+GRU (68.46% accuracy, 0.0249 loss) and VGG16+SimpleRNN (67.20% accuracy, 0.0253 loss). In the case of VGG16+LSTM, the utilization of LSTM can assist the model in capturing more complex relationships within the image data compared to GRU or SimpleRNN.

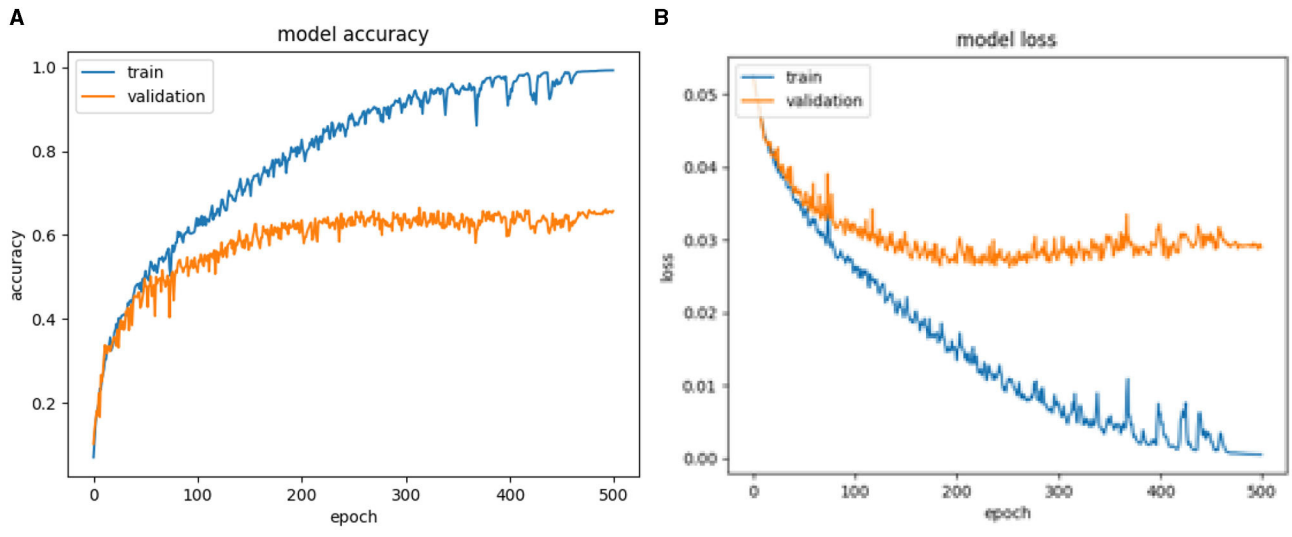

FIGURE 7
The chart describes the accuracy **(A)** and loss **(B)** of improveSORT+GRU algorithm.

Based on Figures 9, 10, as well as Table 2, the SORT+LSTM and SORT+GRU algorithms exhibit lower accuracy in comparison to other models, achieving accuracies of 61.05 and 58.90%, respectively. These algorithms appear to underperform in the task of violence classification. In contrast, the proposed algorithm, improved SORT+LSTM, achieves the highest accuracy at 72.13% with the lowest loss of 0.0221. This algorithm demonstrates significant improvements when compared to both VGG16-based architectures and the original SORT+LSTM algorithm, as shown in Figure 8 and Table 2.

The improveSORT+GRU algorithm achieves an accuracy of 66.14% with a loss of 0.0291. While it outperforms SORT+GRU, it still falls short in comparison to the improved SORT+LSTM and VGG16+LSTM architectures as illustrated in Figure 7 and Table 2. In summary, the improved SORT+LSTM and improved SORT+GRU algorithms attain accuracies of 72.13 and 66.14%, respectively, which surpass the SORT+LSTM and SORT+GRU algorithms with accuracies of 61.05 and 58.9%, respectively. These results suggest that object localization and recovery techniques effectively enhance the accuracy of violent classification algorithms.

Furthermore, the improved SORT+LSTM algorithm, which is based on the SORT algorithm, performs slightly better than the VGG16+LSTM algorithm with an accuracy of 71.11%, the VGG16+SimpleRNN algorithm with an accuracy of 67.2%, and the VGG16+GRU algorithm with an accuracy of 68.46%. Overall, the results presented in the argument support the use of object localization and recovery techniques as an effective approach to improving the accuracy of violence classification.

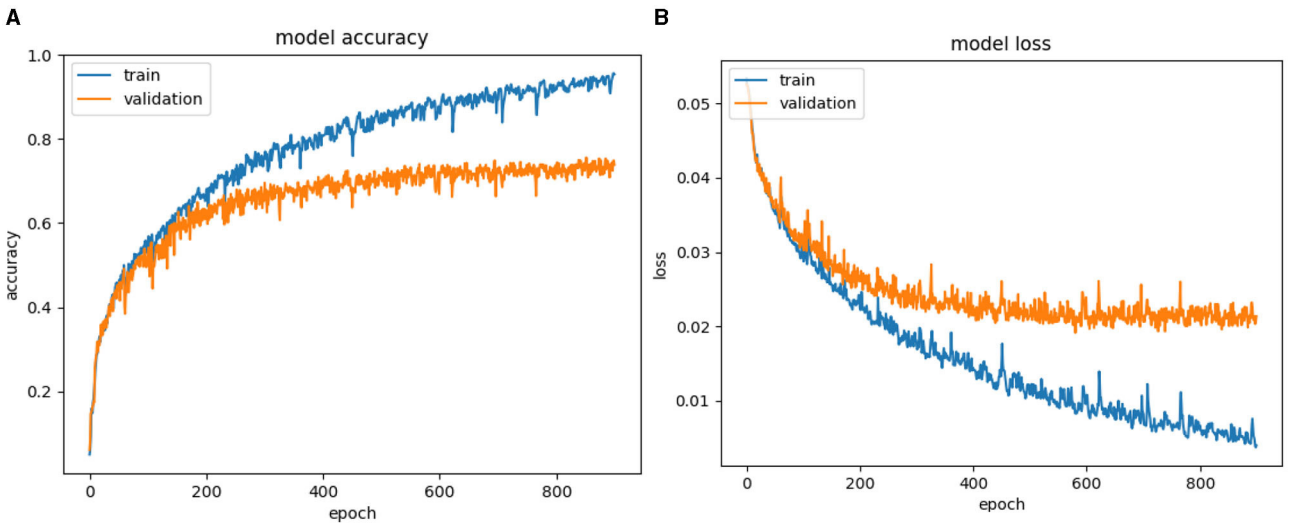

FIGURE 8
The chart describes the accuracy **(A)** and loss **(B)** of improveSORT+LSTM algorithm.

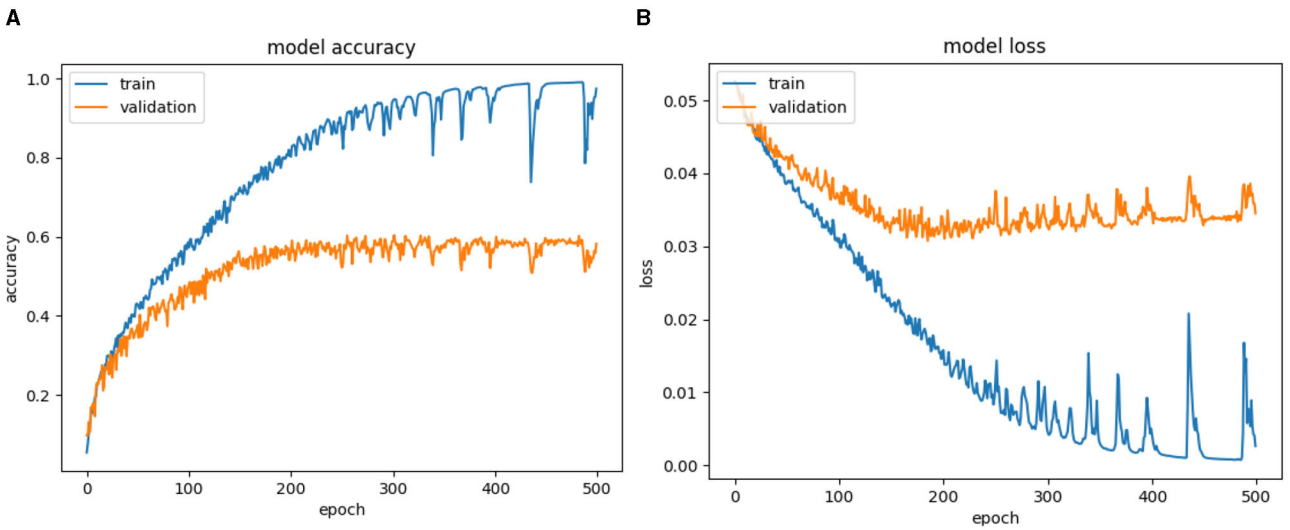

FIGURE 9
The chart describes the accuracy **(A)** and loss **(B)** of SORT+GRU algorithm.

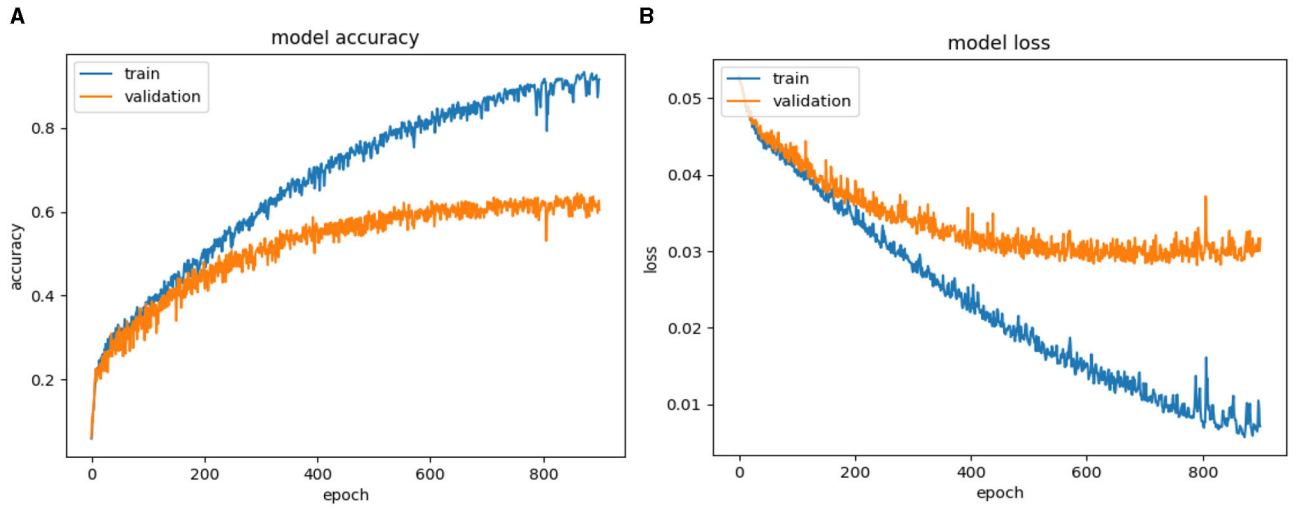

FIGURE 10
The chart describes the accuracy **(A)** and loss **(B)** of SORT+LSTM algorithm.

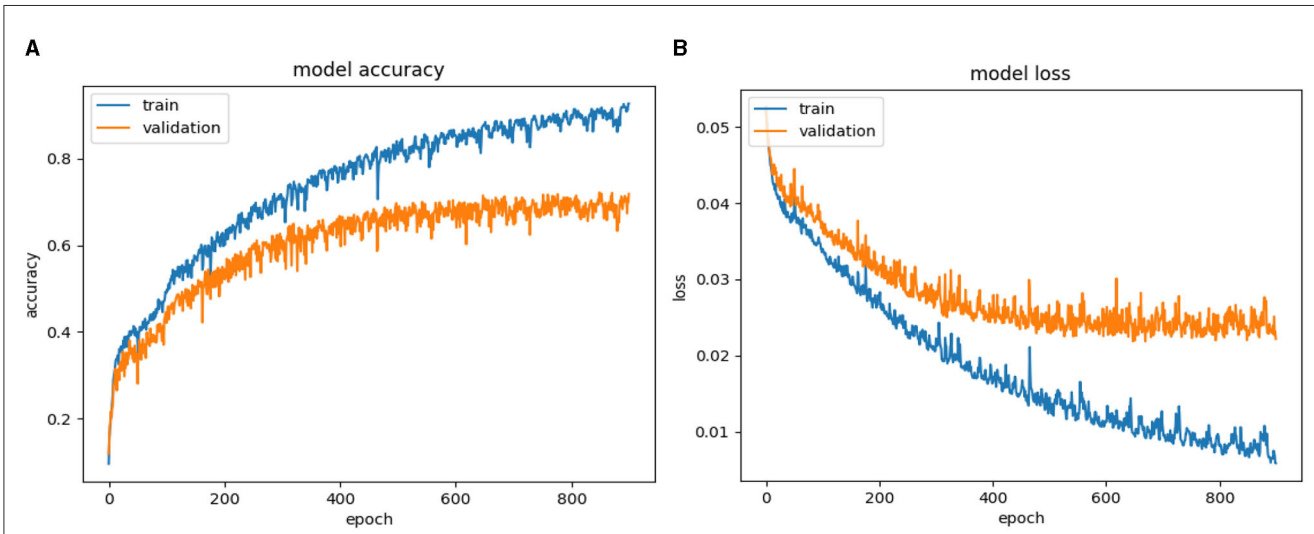

FIGURE 11
The chart describes the accuracy **(A)** and loss **(B)** of VGG16+LSTM algorithm.

We extracted N equally spaced frames from each video and resized them to 224 × 224 dimensions. On average, each frame, after passing through the improved SORT+VGG+LSTM algorithm, is processed and classified within 0.33 s. Specifically, we first use the improved SORT model to eliminate the background from the frame. Subsequently, we utilize this frame to extract features using the VGG16 model. These features are then fed into a fully connected layer for classification, followed by a sigmoid output layer. The fully connected layers employ ReLU activation functions. Additionally, we designed an LSTM layer with 128 LSTM units. The optimization algorithm for the models is Adam, and the loss function used is binary cross-entropy.

5. Conclusion

The article introduced various techniques for classifying violence and tracking objects in videos, specifically when tracking humans in videos with a relatively stable camera and rotational angles that maintain the proportions of the human body, we employed the SORT algorithm. This algorithm enables object tracking, combined with object localization and recovery over a specific time interval. The experimental results show promise in educational settings, where the assumption of camera stability is justified. This distinction is crucial because violence in educational environments exhibits unique characteristics, setting it apart from other forms of violence. In the near future, we plan to evaluate the

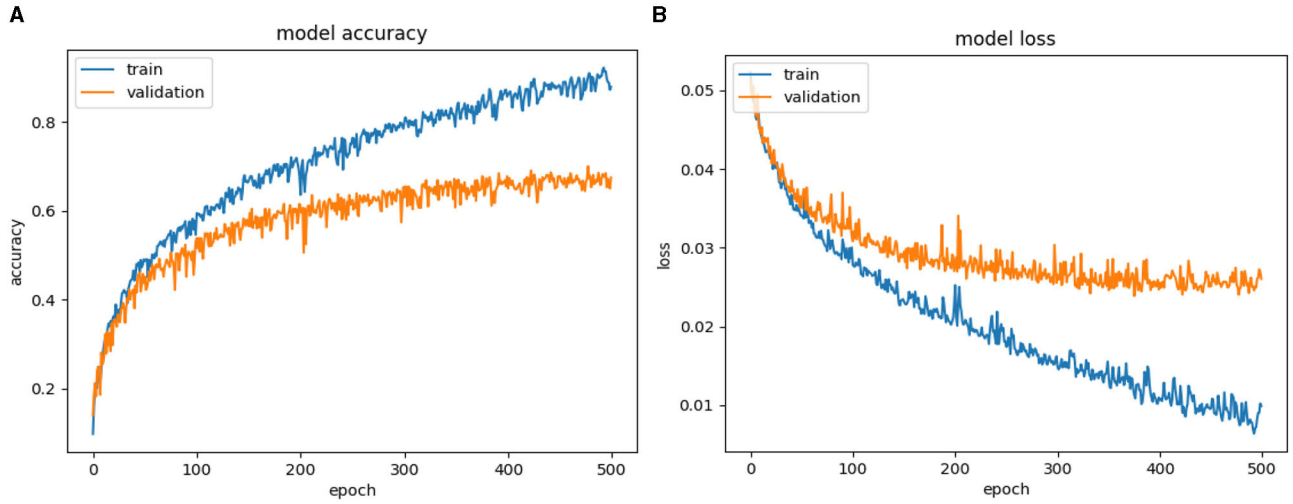

FIGURE 12
The chart describes the accuracy **(A)** and loss **(B)** of VGG16+SimpleRNN algorithm.

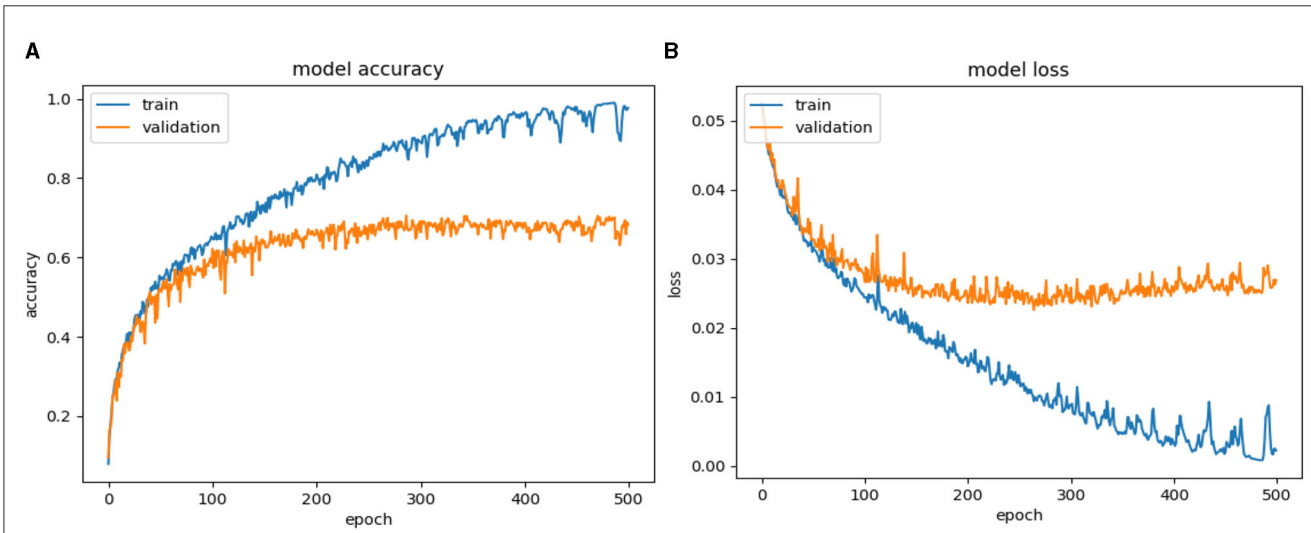

FIGURE 13
The chart describes the accuracy **(A)** and loss **(B)** of VGG16+GRU algorithm.

TABLE 2 **Accuracy and loss of test set between algorithms.**

Algorithm	Accuracy	Loss
VGG16+GRU	68.46%	0.0249
VGG16+SimpleRNN (Zhu and Chollet, 2015)	67.20%	0.0253
VGG16+LSTM	71.11%	0.0230
SORT+LSTM	61.05%	0.0309
Improved SORT+LSTM	**72.13%**	**0.0221**
SORT+GRU	58.90%	0.0343
Improved SORT+GRU	66.14%	0.0291

The bolded row is the highest level of accuracy.

Deep SORT method in conjunction with recurrent neural networks and integrate additional object tracking techniques to provide a more robust and versatile solution capable of handling a wider range of video scenarios while accommodating the unique demands of violence detection in educational contexts.

Because the problem uses the method of object localization, the model is effective at recognizing pre-trained background frames; even with a new background or new space, the model does not reduce accuracy.

To improve the accuracy of school violence classification, in the future, we need to improve the efficiency of object detection and tracking to increase the accuracy of object localization.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

NH: Writing—original draft. NT: Writing—review and editing. LT: Writing—review and editing. IS: Writing—review and editing. PB: Writing—review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Anh, T. Q., Bao, P. T., Khanh, T. T., Thao, B. N. D., Tuan, T. A., and Nhut, N. T. (2012). Video retrieval using histogram and sift combined with graph-based image segmentation. *J. Comp. Sci.* 8, 853–858. doi: 10.3844/jcssp.2012.853.858

Bewley, A., Zongyuan, G., Ott, L., Ramos, F., and Upcroft, B. (2016). "Simple online and realtime tracking," in *Proceedings of the 2016 IEEE International Conference on Image Processing* (Phoenix, AZ, USA), 3464–3468.

Bilinski, P., and Bremond, F. (2016). "Human violence recognition and detection in surveillance videos," in *Proceedings of the 13th IEEE International Conference on Advanced Video and Signal Based Surveillance* (Colorado Springs, CO, USA), 30–36.

Biswas, R., Vasan, A., and Roy, S. S. (2020). Dilated deep neural network for segmentation of retinal blood vessels in fundus images, Iranian Journal of Science and Technology. *Trans. Electr. Eng.* 44, 505–518. doi: 10.1007/s40998-019-00213-7

Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv*. doi: 10.3115/v1/D14-1179

Dinesh, J. S. R., Fenil, E., Gunasekaran, M., Vivekananda, G. N., Thanjaivadivel, T., Jeeva, S., et al. (2019). Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. *Comp. Networks.* 151, 191–200. doi: 10.1016/j.comnet.2019.01.028

Fang, Z., Fei, F., Fang, Y., Lee, C., Xiong, N., Shu, L., et al. (2016). Abnormal event detection in crowded scenes based on deep learning. *Multim. Tools Appl.* 75, 14617–14639. doi: 10.1007/s11042-016-3316-3

Gao, Y., Liu, H., Sun, X., Wang, C., and Liu, Y. (2016). Violence detection using oriented violent flows. *Image Vision Comp.* 48, 37–41. doi: 10.1016/j.imavis.2016.01.006

Han, J., Shao, L., Xu, D., and Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: a review. *IEEE Trans. Cybern.* 43, 1318–1334. doi: 10.1109/TCYB.2013.2265378

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huang, J. F., and Chen, S. (2014). "Detection of violent crowd behavior based on statistical characteristics of the optical flow," in *Proceedings of the 11th International Conference on Fuzzy Systems and Knowledge Discovery* (Xiamen, China), 565–569.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *J. Basic Eng.* 82, 35–45.

Kang, M. S., Park, R. H., and Park, H. M. (2021). Efficient spatio-temporal modeling methods for real-time violence recognition. *IEEE Access.* 9, 76270–76285. doi: 10.1109/ACCESS.2021.3083273

Klein, D. A., Schulz, D., Frintrop, S., and Cremers, A. B. (2010). "Adaptive real-time video-tracking for arbitrary objects," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Taipei, Taiwan), 772–777.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.* 2, 83–97. doi: 10.1002/nav.3800020109

Lee, J., Jin, L., Park, D., and Chung, Y. (2016). Automatic recognition of aggressive behavior in pigs using a kinect depth sensor. *Sensors.* 16, 1–11. doi: 10.3390/s16050631

Liu, Z., Wang, X., Wang, C., Liu, W., and Bai, X. (2023). SparseTrack: multi-object tracking by performing scene decomposition based on pseudo-depth. *arXiv* arXiv:2306.05238. doi: 10.48550/arXiv.2306.05238

Mahmoodi, J., and Salajeghe, A. (2019). A classification method based on optical flow for violence detection. *Expert Syst. With Appl.* 127, 121–127. doi: 10.1016/j.eswa.2019.02.032

Naik, A. J., and Gopalakrishna, M. T. (2016). Violence detection in surveillance video-a survey. *Int. J. Latest Res. Eng. Technol.* 11–17. Available online at: http://www.ijlret.com/NC3PS-2016.html

Pang, J. M., Yap, V. V., and Soh, C. S. (2014). "Human behavioral analytics system for video surveillance," in *Proceedings of the 2014 IEEE International Conference on Control System* (Penang, Malaysia), 23–28.

Ramzan, M., Abid, A., Khan, H. U., Awan, S. M., Ismail, A., Ahmed, M., et al. (2019). A. Mahmood, A review on state-of-the-art violence detection techniques. *IEEE Access.* 7, 107560–107575. doi: 10.1109/ACCESS.2019.2932114

Ren, S., He, K., Girshick, R., and Jian Sun, J. (2017). "Faster R-CNN: towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Roy, S. S., Rodrigues, N., and Taguchi, Y. (2020). Incremental dilations using CNN for brain tumor classification. *Appl. Sci.* 10, 915. doi: 10.3390/app10144915

Saif, A. F. M. S., and Mahayuddin, Z. R. (2020). Moment features based violence action detection using optical flow. *Int. J. Adv. Comp. Sci. Appl.* 11, 503–510. doi: 10.14569/IJACSA.2020.0111163

Shehzed, A., Jalal, A., and Kim, K. (2019). "Multi-Person tracking in smart surveillance system for crowd counting and normal/abnormal events detection," in *Proceedings of the 2019 International Conference on Applied and Engineering Mathematics* (Taxila, Pakistan), 163–168.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. doi: 10.48550/arXiv.1409.1556

Souza, F. D. M. D., Chávez, G. C., Valle, E. A. D., Jr., and Araujo, A.D.A. (2010). "Violence detection in video using spatio-temporal features," in *Proceedings of the 23rd SIBGRAPI Conference on Graphics, Patterns and Images* (Gramado, Brazil), 224–230.

Sudhakaran, S., and Lanz, O. (2017). "Learning to detect violent videos using convolutional long short-term memory," in *Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance* (Lecce, Italy), 1–6.

Ullah, F. U. M., Muhammad, K., Haq, L. U., Khan, N., Heidari, A. A., Baik, S. W., et al. (2021). AI assisted edge vision for violence detection in IoT based industrial surveillance networks. *IEEE Trans. Indust. Inform*. 18, 377. doi: 10.1109/TII.2021.3116377

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016). "Temporal segment networks: towards good practices for deep action recognition," *in Proceedings of the 14th European Conference on Computer Vision* (Amsterdam, The Netherlands), 20–36.

Wang, Q., Ma, Q., Luo, C. H., Liu, H. Y., and Zhang, C. L. (2016). Hybrid histogram of oriented optical flow for abnormal behavior detection in crowd scene. *Int. J. Pattern Recog. Artif. Intellig.* 30, 1–14. doi: 10.1142/S0218001416550077

Wen, S., and Liu, J. (2008). "Current status, causes and intervention strategies of soccer violence in chinese professional football league," in *Proceedings of the 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop* (Wuhan, China), 1145–1147.

Ye, L., Liu, T., Han, T., Ferdinando, H., Seppänen, T., and Alasaarela, E. (2021). Campus violence detection based on artificial intelligent interpretation of surveillance video sequences. *Remote Sensing*. 13, 628. doi: 10.3390/rs13040628

Zhang, T., Yang, Z., Jia, W., Yang, B., Yang, J., and He, X. (2015). A new method for violence detection in surveillance scenes. *Multimedia Tools Appl.* 75, 7327–7349. doi: 10.1007/s11042-015-2648-8

Zhou, L. (2022). "End-to-end video violence detection with transformer," in *Proceedings of the 5th International Conference on Pattern Recognition and Artificial Intelligence (IEEE)*, 880–884.

Zhou, P., Ding, Q., Luo, H., and Hou, X. (2017). "Violent interaction detection in video based on deep learning," in *Proceedings of the 6th Conference on Advances in Optoelectronics and Micro/nano-optics* (Nanjing, China).

Zhu, S., and Chollet, F. (2015). *Keras*. Available online at: https://keras.io (accessed September 05, 2022).